CAMERA READY COPY

University of Arizona - Grant

NGR-03-002-011

AN ENDPOINT SUFFICIENCY CONDITION
FOR MULTIPLE STATIONARY SOLUTIONS

BY

Richard G. Brusch and Thomas L. Vincent

prepared by

The University of Arizona
Tucson, Arizona

for

National Aeronautics and Space Administration

August, 1969

TABLE OF CONTENTS

Page

TABLE OF CONTENTS - Cont'd.

TABLE OF CONTENTS - Cont'd.

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS - Cont'd.

LIST OF ILLUSTRATIONS - Cont'd.

ABSTRACT

The existence of multiple solutions to a general Problem of Bolza in the calculus of variations is investigated. These multiple stationary solutions are of several distinct types, each of which is briefly discussed.

The most important type of multiple stationary solutions occurs when a complete set of necessary and sufficient conditions have not been applied. A sufficiency test to eliminate some multiple solutions of this type is developed. It is shown that a sufficiency test can be broken down into a path sufficiency test and an endpoint sufficiency test and that once the first necessary conditions of the calculus of variations have been applied, the path sufficiency test and the endpoint sufficiency test can be applied independently.

Only the endpoint sufficiency test was investigated. Analytical application of this condition requires the analytical integration of a set of nonlinear differential equations subject to mixed boundary conditions. Since this is often difficult or impossible, an algorithm is developed for numerically implementing the endpoint sufficiency condition. A geodetics problem is solved analytically to illustrate the theory and demonstrates that the sufficiency condition is an effective computational tool, by eliminating certain classes of non-optimal solutions from consideration.

A survey of numerical methods for solving differential equations with mixed boundary conditions arising from problems in the calculus of variations is presented. The generalized Newton-Raphson method for solving such problems is developed in detail and digital computer programs for implementing it are included.

The problem of determining minimum fuel, orbital transfers for low-thrust space vehicles is treated as a final comprehensive example. Transfers are considered which begin in an initial circular orbit and terminate in any final elliptic orbit in the same plane. In addition the final argument of periapsis, the final true anomoly, the final range angle, and the final time are all unspecified and considered free. Several examples of multiple stationary solutions to a given terminal orbit are presented. Digital computer programs for solving this problem and for implementing the endpoint sufficiency test are included. The results of a comprehensive study of fixed mass, single thrust, orbital transfers from initial circular orbits to a wide range of final elliptic orbits are presented.

CHAPTER 1

INTRODUCTION

A renewed interest in the theory of the calculus of variations has been witnessed in the past decade due in part to the development of sophisticated automatic control systems. When the concepts of state and control variables from the theory of automatic control systems were incorporated in the theory of the calculus of variations, an entirely new practical viewpoint resulted. Once left completely to the realm of pure mathematicians, the calculus of variations, in its new formulation, speaks a language easily understood by engineers. With the potential use of the calculus of variations made apparent to a much broader spectrum of technology, rapid theoretical development has been inevitable.

Significant advances have been made in casting heretofore diverse calculus of variations problems into a single generalized formulation, e.g., the Problem of Bolza (Bliss, 1946, pp. 187-265). Although not entirely complete, the formulation of the theory of the calculus of variations in control notation used by Vincent and Mason (1969) is applicable to most problems of engineering importance.

A limitation to the engineering effectiveness of the calculus of variations in its current formulation has been the lack of a complete set of sufficiency conditions expressed in control notation. Although a fairly complete set of sufficiency theorems has been developed for the classical problems of the calculus of variations (Bliss, 1946, pp. 235-265), they are formulated in a manner and in a mathematical notation which is difficult to apply to most practical problems. For the most general problems expressed in control notation, a complete set of sufficiency conditions is not known. The lack of such a comprehensive engineering formulation has placed heavy emphasis on the more easily understood necessary conditions.

Trajectories which satisfy all of the necessary conditions of the calculus of variations are referred to as stationary solutions. Stationary solutions are often accepted as optimum trajectories without further investigation. For most elementary problems this practice is adequate. However, stationary solutions are only candidates for the true optimum. When an investigator obtains two or more distinct stationary solutions to his problem, this point becomes quite clear. Chapter 2 partially resolves the problem of multiple stationary solutions by developing a sufficiency condition in modern control notation for problems with variable endpoints. This sufficiency condition is related to classical sufficiency conditions. In addition, a numerical algorithm will be presented which allows an investigator to apply the endpoint sufficiency condition in cases where the differential equations describing the optimum trajectory cannot be integrated analytically.

However complete a theory may be, it is of little use to the engineer unless it leads in a direct fashion to solutions of real-world problems. Unfortunately, the theory of the calculus of variations inevitably requires the integration of a set of nonlinear ordinary differential equations with mixed boundary conditions. In all but the simplest academic problems, analytic solutions cannot be obtained. In order for the calculus of variations to be a useful engineering tool, a practical computational algorithm must be developed. The problem of finding numerical solutions to differential equations with mixed boundary conditions is the subject of Chapter 3.

Chapter 4 presents an interesting example problem in which bounded control gives rise to multiple stationary solutions in a unique fashion.

The complex problem of optimum low-thrust orbital transfers is investigated in Chapter 5. Several types of multiple stationary solutions are exhibited and the end-point sufficiency condition is used to distinguish the true optimum from among the candidates. In addition, the results of a comprehensive study of minimum time low-thrust orbital transfers is presented. The investigation concerns the qualitative aspects of transfers from an initial circular orbit to any given final elliptic orbit with the angle of transfer and relative argument of periapsis unspecified.

1.1 Historical Background

The historical development of the calculus of variations has been sporadic, and not entirely free of controversy. Without going into the details of the early development, it is appropriate to mention the outstanding contributions and references of historical importance.

In 1696 the Brachistochrone problem was proposed by John Bernoulli (Ostwald, 1911, no. 46, p. 3). He sought the path which minimized the time required for a mass under the influence of gravity to slide without friction from one point to another. In 1744, Euler (1911) discovered the characteristic differential equation which must be satisfied by optimum trajectories. Lagrange analytically formalized the work of Euler and extended the scope of problems to two independent variables. While Euler and Lagrange considered necessary conditions for a stationary curve, Legendre (1786), through the use of the so-called second variation, developed a further necessary condition that distinguished maximal extremals from minimal extremals. Jacobi (1837) constructed an essential modification of the Legendre necessary condition, and from it deduced a further test, which set limits on the range of the independent variable. Clebsch (1858) and Mayer (1868) conducted further investigations connected with transformations of the second variation.

Much of modern theory of the calculus of variations is based upon proofs employing strong variations which were first developed by Weierstrass. The work of Weierstrass, although never formally published, is well known largely through the publications of his contemporaries; among them, the works of Bolza (1913), Kneser (1900), Forsyth (1927) and Hilbert (1902) have had the most lasting influence. The work of Bolza has been of particular influence on current investigators. He formulated a general problem, now known as the Problem of Bolza, which included the problems of Lagrange and Mayer as special cases (Bliss, 1946, pp. 189-193). A considerable interest was taken in the Problem of Bolza, and a summary of the works of many investigators is presented in the comprehensive book by G. A. Bliss (1946). Recently

Hestenes (1966) has reformulated the Problem of Bolza from the classical dependent variables notation to the modern, control variable, state variable notation. He obtains only path sufficiency theorems for problems with fixed endpoints and does not discuss the sufficiency condition for problems with variable endpoints to be developed in Chapter 2.

1.2 The Problem of Bolza

Since much of what is to follow depends upon an understanding of the Problem of Bolza as formulated in control notation (Vincent and Brusch, 1966, pp. 4-5), a brief statement of the problem in its simplest form is appropriate.

Among the set of all continuous state functions,

$$y_i(t) \qquad i = 1,2,\ldots,n;\ t_o \leq t \leq t_f \tag{1.2.1}$$

and continuous control variable functions

$$u_k(t) \qquad k = 1,2,\ldots,\ m < n \tag{1.2.2}$$

satisfying differential equations and end-conditions of the form

$$\dot{y}_i = f_i(y_j,u_k,t) \quad j = 1,2,\ldots,n \tag{1.2.3}$$

$$\psi_\ell(y_{io},y_{if},t_o,t_f) = 0$$
$$\ell = 1,2,\ldots,p \leq 2n + 2 \tag{1.2.4}$$

find the set which will minimize a sum of the form:

$$J' = g(y_{io}, y_{if}, t_o, t_f) + \int_{t_o}^{t_f} L(y_i(t), u_k(t), t)dt. \quad (1.2.5)$$

Here it is assumed that the functions f_j, L, g, and ψ_ℓ are of class C^2. In the above and throughout this presentation, a dot above a variable will be used to represent the derivative of the variable with respect to t, the independent variable. Likewise the subscripts o and f will indicate the evaluation of the variable or expression at the initial and final value of t, respectively. For the sake of brevity, the range of subscripts i, j, k, and ℓ will be as given above and will not be repeated in what follows.

Following the conventional method of Lagrange multipliers (Bryson and Ho, 1969), minimization of the augmented function

$$J^* = g + \mu_\ell \psi_\ell + \int_{t_o}^{t_f} [L - \lambda_i f_i + \lambda_i y_i]dt. \quad (1.2.6)$$

is considered. In the above equation and throughout the presentation, repeated subscripts will be used to imply summation. Equation (1.2.6) was obtained by adjoining equations (1.2.3) and (1.2.4) to relation (1.2.5) as follows:

(a) multiplying relations (1.2.3) by the variables $\lambda_i(t)$, respectively, integrating from

t_o to t_f and by adding the sum of the integrals to expression (1.2.5),

(b) multiplying equations (1.2.4) by the parameters μ_ℓ and adding the sum of the products to expression (1.25).

It is convenient to define the following functions:

$$G(y_{io}, y_{if}, t_o, t_f) = g + \mu_\ell \psi_\ell \qquad (1.2.7)$$

$$H[y_i(t), \lambda_i(t), u_k(t), t] = \lambda_i f_i - L \qquad (1.2.8)$$

The function H is often referred to as the Hamiltonian. With these definitions, equation (1.2.6) may be written as

$$J^* = G + \int_{t_o}^{t_f} [-H + \lambda_i \dot{y}_i] dt \qquad (1.2.9)$$

By considering small variations in the path and endpoints about a nominal path, it can be shown that if the functions $u_k(t)$ and $y_i(t)$ are a solution to the Problem of Bolza, then they must satisfy the following necessary conditions (Hestenes, 1966, pp. 346-351):

Condition I. There exist continuous multipliers $\lambda_i(t)$ and Hamiltonian function as defined in equation (1.2.8) such that:

(1) the Euler-Lagrange equations,

$$\dot{\lambda}_i = - \frac{\partial H}{\partial y_i} \qquad (1.2.10)$$

$$\frac{\partial H}{\partial u_k} = 0 \qquad (1.2.11)$$

are satisfied at every point along the path and,

(2) the transversality conditions

$$\frac{\partial G}{\partial t_o} + H_o = 0 \qquad (1.2.12)$$

$$\frac{\partial G}{\partial y_{io}} - \lambda_{io} = 0 \qquad (1.2.13)$$

$$\frac{\partial G}{\partial t_f} - H_f = 0 \qquad (1.2.14)$$

$$\frac{\partial G}{\partial y_{if}} + \lambda_{if} = 0 \qquad (1.2.15)$$

are satisfied by the endpoints.

Condition II. The inequality

$$H[y_i(t),\lambda_i(t),u_k^{no}(t),t] \leq H[y_i(t),\lambda_i(t),u_k(t),t] \qquad (1.2.16)$$

must be satisfied for all $t_o \leq t \leq t_f$ and for all non-optimal control functions u^{no}. This is referred to as the necessary condition of Weierstrass.

Condition III. The k by k matrix

$$\frac{\partial^2 H}{\partial u_s \partial u_t} \qquad \begin{array}{l} s = 1,2,\ldots,k \\ t = 1,2,\ldots,k \end{array} \qquad (1.2.17)$$

must be negative semi-definite for a minimum. This condition is known as the Legendre-Clebsch necessary condition.

Condition IV. A fourth necessary condition is discussed by Bliss (1946, pp. 226-228) in classical dependent variable notation. He proves that the second order variation of a sum similar to J^* must be non-negative along a stationary arc, if that arc minimizes J^*.

Hestenes (1966, pp. 283-286) verifies this conclusion in modern control notation for the Problem of Bolza with fixed endpoints. As developed by Hestenes, the fourth necessary condition represents a necessary condition on the path alone; variations in the endpoints are not considered. If Condition III is satisfied, Condition IV is usually referred to as the Jacobi Condition. A further geometric interpretation of this condition is presented in the following section.

1.3 The Nature of Multiple Stationary Solutions

It is helpful in understanding the nature of multiple stationary solutions to classify them by the circumstances pertinent to their occurrence. The classification is not an apriori one, but rather one based on experience.

1.3.1 Fixed Endpoint Problems and Path Sufficiency Conditions.

Multiple stationary solutions are often obtained for problems with fixed endpoints, because only the first three necessary conditions have been considered. Bliss (1946, p. 235) has shown in dependent variable notation that the fourth necessary condition of Jacobi, taken together with the first three necessary conditions, suitably strengthened, forms a sufficient set of conditions for the Problem of Bolza.

Although analytically complex, the Jacobi condition has a simple geometric interpretation for problems with one state variable. In this case the set of all solutions forms a one dimensional family of extremal arcs, y = y(t,c), which all pass through the initial point as shown in Figure (1.1). With each arc is associated a particular value of c. If arcs y(t,c) and y(t,c + ϵ) intersect in the limit as ϵ goes to 0, the point of intersection is called a conjugate point. The locus of such intersections is called the discriminant locus, also shown in Figure (1.1).

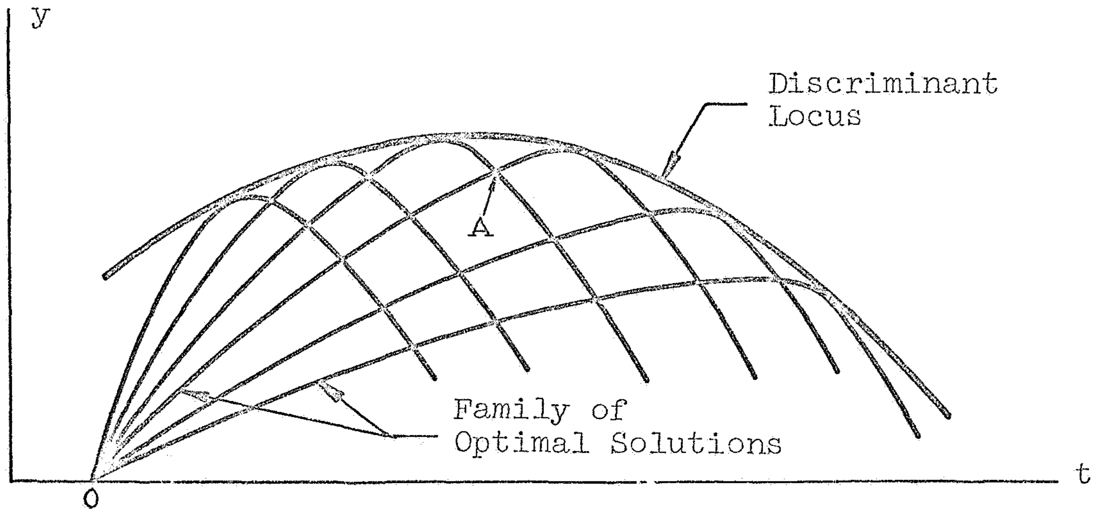

Fig. 1.1 A Family of Extremals and the Discriminant Locus

In terms of this geometry, the Jacobi condition requires that an optimal trajectory contain no conjugate point. Alternatively, the condition requires that an optimal solution may not touch the discriminant locus. Figure (1.1) shows that there are two solutions joining points O and A, one of which touches the discriminant locus and is therefore non-optimal. If the fourth necessary condition of Jacobi is applied in such cases of multiple stationary solutions, all but one of the trajectories will be shown to contain a point conjugate to the initial point, thus rendering them non-optimal.

Examples of this occurrence are many. The Brachistochrone problem with fixed endpoints graphically illustrates the problem. Consider the problem of a bead sliding down a wire under the influence of gravity alone. What should the shape of the wire be in order to minimize the time of transit between two points in a vertical plane? It is well known that the solution curves are cycloids. However, as shown in Figure 1.2, there are several different cycloids which satisfy the necessary conditions of the calculus of variations. It can be seen that the x-axis forms the discriminant locus and that the points where solutions 1 and 2 touch the discriminant locus are conjugate points. Since solutions 1 and 2 violate the Jacobi condition, it is evident that solution 3 is the true optimum.

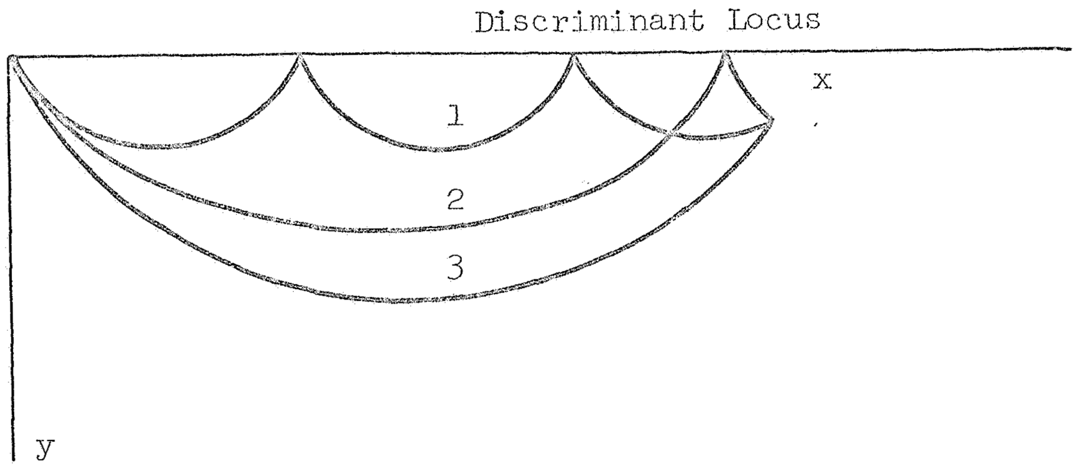

Fig. 1.2 Multiple Stationary Solutions for the Brachistochrone Problem

In this case it has been possible to distinguish the true optimum from the candidates by applying sufficiency conditions pertaining to the path. Although the fourth necessary condition of Jacobi has been exhibited in control notation for fixed endpoint problems (Hestenes, 1966, pp. 250-286), it has not been frequently used in engineering problems due to complications in applying it to cases where the Euler-Lagrange differential equations cannot be integrated analytically.

1.3.2 Variable Endpoint Problems. Problems with variable endpoints require that the endpoints of the trajectories, as well as the path, be selected in an optimum fashion. Consider the geodesics problem of trying to find the minimum distance from the origin to a given parabola as shown in Figure (1.3). It will be shown in Chapter 2 that two stationary solutions exist, viz., OA and OB. Once an endpoint is determined, the problem becomes one of fixed endpoints, and the path sufficiency conditions previously discussed can then be applied. In this case, with the endpoint, A or B, thought of as being fixed, it can be shown that both solutions satisfy the Jacobi sufficiency conditions regarding path (Bolza, 1961, pp. 84-86). Since both paths satisfy sufficiency conditions, but the endpoints were determined from necessary

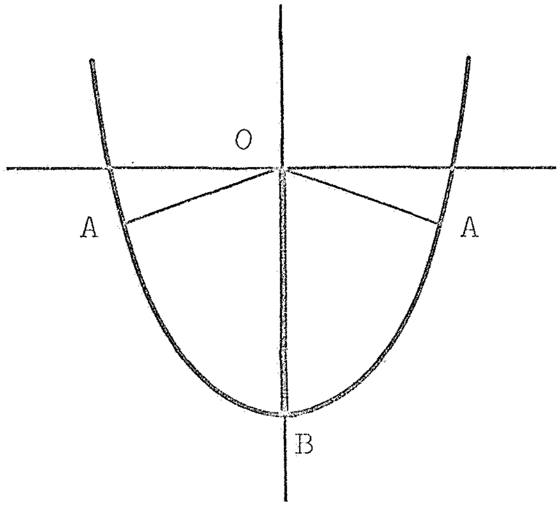

Fig. 1.3 Multiple Stationary Solutions to a Geodetics Problem

conditions only, it is apparent that a sufficiency condition regarding endpoints is needed to distinguish the true optimal. Such an endpoint sufficiency condition does exist and will be derived in Chapter 2. It will be shown that only one of the solutions will represent a local minimum with respect to variation of the endpoint along the parabola.

1.3.3 Problems Requiring Path and Endpoint Sufficiency Conditions. In the last two sections, the necessity of using path sufficiency conditions and endpoint sufficiency conditions was illustrated separately. It is not unusual, however, to encounter problems requiring the application of both sufficiency conditions to distinguish the true optimum from the set of multiple stationary solutions. To illustrate this situation, reconsider the Brachistrochrone problem where the final endpoint, instead of being fixed, is required to be on curve E as shown in Figure (1.4). Both trajectories OAB and OB satisfy the endpoint sufficiency conditions; that is, both solutions represent a local minimum with respect to small variations of the endpoint along endpoint manifold E. As discussed before, point A is a conjugate point, thus violating the Jacobi path sufficiency condition. Trajectory OC violates the endpoint point sufficiency condition; that is, endpoint C represents a local maximum with respect to

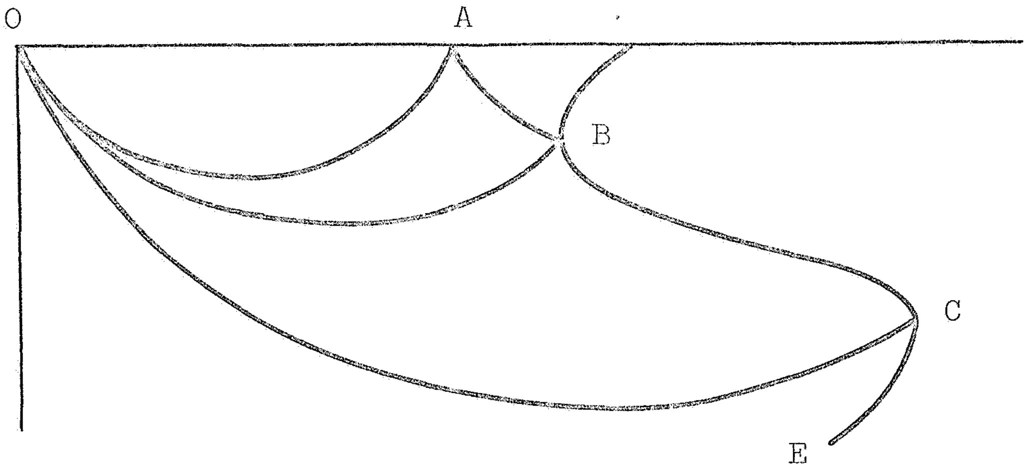

Fig. 1.4 A Problem Requiring both Path and Endpoint Sufficiency Conditions

small variations of the endpoint along E. Thus through the use of both sufficiency conditions, OB is selected as the true optimum.

1.3.4 Problems with Periodic Solutions. Consider a system of equations (1.2.3) which exhibit periodic oscillations when no control effort is applied. It is not unusual for the optimal controls and adjoint variables of such a system to also demonstrate periodic motion with the same period. This is especially true if the magnitude of the control is small. By restricting the magnitude of the control to sufficiently small values, the deviation between the solution from one period to the next can be made as small as desired. Problems with bounded control often have controls with such small magnitudes.

The criterion by which the solution is terminated is often periodic for problems exhibiting periodic oscillations. The terminating or cutoff condition is obtained from the transversality conditions (1.2.12) - (1.2.15) by eliminating the μ_ℓ parameters, to form a single relationship among the state and adjoint variables. The zeros of the cutoff function then represent the terminating condition. As an example, consider the problem of a thrusting harmonic oscillator: a mass is connected in parallel by a spring and dashpot to an inertial reference. The mass is capable of generating a bounded thrust in the upward direction. The problem is to get the mass to a specified height while minimizing the integral of the thrust with respect to time. For simplicity it is assumed that the mass is constant. This problem is discussed in detail in Chapter 4. For small damping factors and null thrust, the state variables, position and velocity, the adjoint variables, and the cutoff function all exhibit damped periodic oscillations. As shown in Figure (1.5) the cutoff condition is satisfied during each period. For sufficiently small thrust amplitudes the cutoff function will deviate only slightly from that generated for null thrust, and will be satisfied at several points.

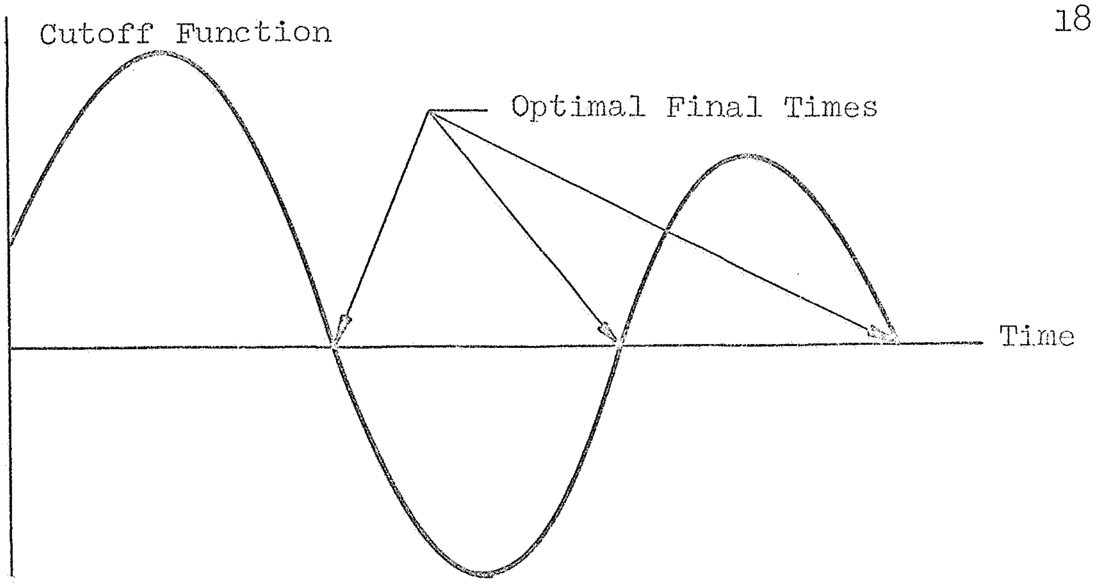

Fig. 1.5 Multiple Solutions Due to a Periodic Cutoff Function

Each time the cutoff condition is satisfied, a potential optimal endpoint and a corresponding stationary solution is obtained. Thus in periodic systems with weak bounded control, multiple stationary solutions may be encountered. Examples of such systems are presented in Chapter 4 and Chapter 5.

1.3.5 Problems with Singular Control. In the case of optimization problems with bounded control variables, for example, $u_{min} \geq u \geq u_{max}$, one encounters the intriguing problem of the possibility of singular control. In cases where the Hamiltonian is linear in a bounded control variable, the control cannot be determined from

Euler-Lagrange equation (1.2.11). In this case the Hamiltonian can be written as

$$H = S(y_i,\lambda_i,t)u + Q(y_i,\lambda_i,t) \qquad (1.6.1)$$

for scaler control. S is referred to as the switching function. The well-known Maximum Principle for problems with bounded control developed by Pontryagin _et_ _al_. (1962) requires that

$$\begin{aligned} u &= u_{max} \quad \text{when } S > 0 \\ u &= u_{min} \quad \text{when } S < 0 \end{aligned} \qquad (1.6.2)$$

However, for the case when $S \equiv 0$ over a non-vanishing time interval, the Maximum Principle is indeterminant, and u may take on intermediate values. This is the case of singular control. George Leitmann has pointed out that (1966, pp. 57-58), "While it is possible in a particular problem...to rule out the possibility of [singular control], this cannot be done in general." Thus, whenever the switching function goes to zero, the control will change, i.e., the control will switch to the opposite extreme or to singular control.

To demonstrate the existence of multiple stationary solutions in the case of singular control, examine the problem of minimizing

$$J = \frac{1}{2}\int_0^\infty x_1^2 \, dt$$

subject to the constraints:

$$\begin{aligned} &\dot{x}_1 = x_2 + u && x_1(0) = x_{10} && x_1(\infty) = 0 \\ &\dot{x}_2 = - u && x_2(0) = x_{20} && x_2(\infty) = 0 \\ &|u| \leq 1 \end{aligned} \qquad (1.6.4)$$

This problem was first discussed by Johnson and Gibson (1963).

The Hamiltonian is

$$H = (\lambda_1-\lambda_2)u + \lambda_1 x_2 - x_1^2/2 = \lambda_1(x_2 + u) - \lambda_2(u) - x_1^2/2$$

In this case $S = \lambda_1 - \lambda_2$. If S is identically zero for a non-vanishing time interval, singular control exists. By taking a suitable number of time derivatives, it can be shown that the singular control is given by $u = -x_1-x_2$, and that the singular arcs are two lines $x_1(t) = 0$ and $x_1(t) + 2x_2(t) = 0$.

Figure (1.6) shows two possible stationary solutions to the problem starting at point A, one of which has a singular subarc. The first arc AB is the same for both solutions. In both solutions u = -1 on arc AB. However, at point B, two choices can be made for the optimal control. One can continue with u = -1 along arc BC and then follow arc CO to the origin with u = + 1. This is the so-called "bang-bang" solution. Alternately, at point B one may elect singular control, $u = x_2$, and proceed to the origin directly along arc BO.

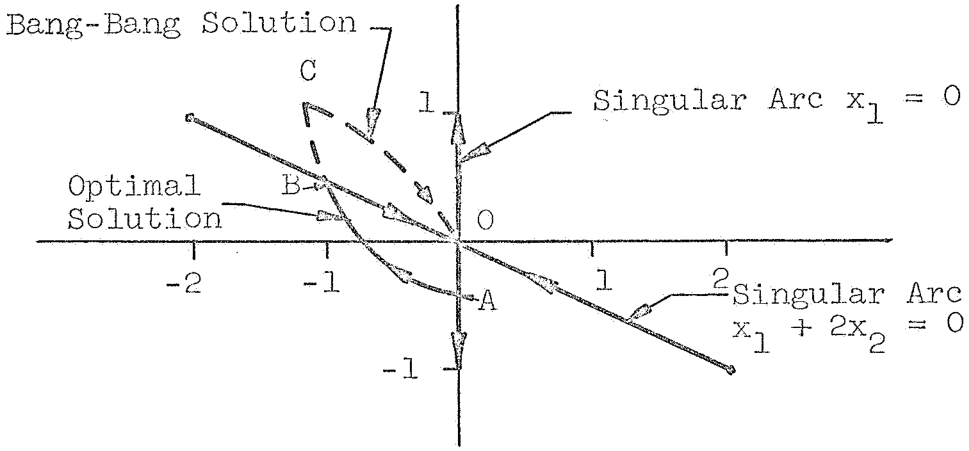

Fig. 1.6 Multiple Solutions Arising from Singular Control

Unfortunately, there is no guarantee that the solution with singular control is minimizing or that it will always enter the optimal solution, even if the possibility of singular solutions does exist. In this case the solution with bang-bang control, arc ABCO, has an index of performance almost 12 per cent larger than for the true optimal control which uses the singular control arc BO.

Recently, Kelly, Kopp and Moyer (1967) and Robbins (1965) have developed a new necessary condition for testing the optimality of singular subarcs. Although this necessary condition will certainly eliminate a substantial number of non-optimal singular solutions, a complete set of necessary and sufficient conditions still remain to be found.

CHAPTER 2

AN ENDPOINT SUFFICIENCY CONDITION

In Chapter 1 it has been shown that failure to apply sufficiency conditions to problems in the calculus of variations often results in multiple solutions, some of which are non-optimal. The examples have established that multiple solutions occur in many types of variational problems and for the simplest of problem formulations. In addition, a complete set of sufficiency conditions needed to select the true optimal has not been formulated in modern state-variable, control-variable notation.

In this chapter an endpoint sufficiency condition is developed for variational problems with variable endpoints. Among the multiple stationary solutions that may exist, the condition provides a test for distinguishing those stationary solution endpoints which represent a minimum. By way of proving the endpoint sufficiency condition, a novel proof of the transversality necessary condition is also exhibited. The endpoint sufficiency condition is related to sufficiency conditions of the classical calculus of variations in section 2.4. In section 2.5 the sufficiency condition is illustrated with an example problem.

Finally, a numerical algorithm is developed for applying the endpoint sufficiency condition to problems with no analytic solution.

2.1 Functional Relationships for the Problem of Bolza

Heuristic arguments in Chapter 1 have implied that the necessary and sufficient conditions for the Problem of Bolza fall into two classes: those pertaining to the path and those pertaining to the endpoints. It has been further implied that sufficiency conditions pertaining to the endpoints can be considered independently from those pertaining to path. Consider the Problem of Bolza as expressed in section 1.2, equations (1.2.1) - (1.2.5). In this formulation, and for the remainder of this chapter, the controls u_k are assumed to be unbounded functions of time. In addition, it is now assumed that the Jacobian is not equal to zero

$$\frac{\partial\left(\frac{\partial H}{\partial u_1}, \frac{\partial H}{\partial u_2}, \ldots, \frac{\partial H}{\partial u_k}\right)}{\partial(u_1,u_2,\ldots,u_k)} \neq 0 \qquad (2.1.1)$$

for all points $(u_1,u_2,\ldots,u_k)$ in the control space. The function H has been previously defined in equation (1.2.8). If equation (2.1.1) is valid, the implicit function theorem (Buck, 1965, pp. 283-286) assures the existence of the k functional relations

$$u_k = u_k[y_i(t),\lambda_i(t)] \qquad (2.1.2)$$

because of the k control variable Euler-Lagrange equations (1.2.11). Condition (2.1.1) specifically eliminates from consideration those systems in which any state variable derivative, $\dot{y}$ as defined in equation (1.2.3), is a linear function of any of the control variables. It is therefore assured that the control variable Euler-Lagrange equations (1.2.11) will be explicit functions of u_k.

A solution to the Problem of Bolza is specified by solutions for the state variables y_i, as well as the control variables u_k, as functions of time. A selection of the initial time and the final time completes the solution. To obtain these solutions, the control variable Euler-Lagrange equations are first solved for the control variables u_k as functions of the state variables y_i and the adjoint variables λ_i. The control variables in the L function in equation (1.2.5) and in the f_i functions in the state variable differential equations (1.2.3) are then replaced by the functional relationship for u given in equation (2.1.2). The functions f_i and the L function are now explicitly dependent only on the state variables, the adjoint variables, and time.

$$f_i = f_i[y_j(t);\ u_k(y_j(t),\lambda_j(t));t] \qquad (2.1.3)$$

$$L = L[y_j(t);u_k(y_j(t),\lambda_j(t));t] \qquad (2.1.4)$$

With relations (2.1.3) and (2.1.4) substituted into the H function, equation (1.2.8), the H function becomes an explicit function of the state variables, the adjoint variables, and time, alone:

$$H^o = H^o[y_i(t);\lambda_i(t);u_k(y_i(t),\lambda_i(t));t] \qquad (2.1.5)$$

Finally, by considering the adjoint variable Euler-Lagrange equation (1.2.10) in view of equation (2.1.5), it is evident that a similar functional relationship exists for the time derivatives of adjoint variables,

$$\dot{\lambda}_i = P_i[y_i(t),\lambda_i(t),t] = -\frac{\partial H}{\partial y_i} \qquad (2.1.6)$$

In summary, once the optimal control is selected, the state variable differential equations (1.2.3) and the adjoint variable differential equations (1.2.10) comprise a set of 2n first order nonlinear differential equations in the 2n state and adjoint variables and time. This set of differential equations can be integrated in theory, yielding

$$y_i = y_i(t,c_r) \qquad r = 1,2,\ldots,2n \qquad (2.1.7)$$

$$\lambda_i = \lambda_i(t,c_r) \qquad (2.1.8)$$

where the c_r's are constants of integration. The p state variable constraints (1.2.4) and the 2n + 2 equations representing the transversality necessary conditions (1.2.12) - (1.2.15) comprise a set of (2n + p + 2) nonlinear algebraic equations in the 2n constants c_r, the p parameters μ_ℓ, the initial time t_o, and the final time t_f. These equations may be solved for the c_r's, t_o and t_f. However, since the set of equations is nonlinear, a unique solution for these quantities is not guaranteed.

The initial values $(y_{io},\lambda_{io},t_o)$ and the final values $(y_{if},\lambda_{if},t_f)$ are specified. Hence, the c_r's may be determined as a function of initial and/or final values by evaluating equations (2.1.7) and (2.1.8) at either the initial or final point. For example, a solution for the c_r's would be specified by the set $(y_{io},\lambda_{io},t_o,t_f)$, the set $(y_{if},\lambda_{if},t_f,t_o)$ or the set (y_{io},y_{if},t_o,t_f). While it is difficult to attach any physical meaning to the initial or final values of the Lagrange multipliers, the initial and final values of the state variables have an immediate physical significance. For this reason, the state variable endpoints have been selected to functionally represent the c_r constants of integration for the rest of this chapter. Thus equations (2.1.7) and (2.1.8) can be written as

$$y_j = y_j(t,y_{io},y_{if},t_o,t_f) \qquad (2.1.9)$$

and $$t_o < t < t_f$$

$$\lambda_j = \lambda_j(t,y_{io},y_{if},t_o,t_f) \qquad (2.1.10)$$

By substituting the functional relationships exhibited in equations (2.1.9) and (2.1.10) into relations (2.1.2) - (2.1.6), it can be seen that the functions u_k,L,f_i,H^o, and P_i can all be written as explicit functions of the set $(t,y_{io},y_{if},t_o,t_f)$. These functional relations, together with that for the function G from equation (1.2.7) are summarized for reference below:

$$u_k = u_k\ (t,y_{io},y_{if},t_o,t_f) \tag{2.1.11}$$

$$L = L\ (t,y_{io},y_{if},t_o,t_f) \tag{2.1.12}$$

$$\dot{y}_j = f_j(t,y_{io},y_{if},t_o,t_f) \tag{2.1.13}$$

$$\dot{\lambda}_j = P_j(t,y_{io},y_{if},t_o,t_f) \tag{2.1.14}$$

$$H^o = H^o[y_j(t,y_{io},y_{if},t_o,t_f);\ \lambda_j(t,y_{io},y_{if},t_o,t_f);\ u_k(t,y_{io},y_{if},t_o,t_f);\ t] \tag{2.1.15}$$

$$G = G(\mu_\ell,y_{io},y_{if},t_o,t_f) \tag{2.1.16}$$

So that there will be no confusion as to the meaning of the subscripts, note that

$$y_{jo} \equiv y_j\Big|_{t=t_o} \tag{2.1.17}$$

$$\lambda_{jo} \equiv \lambda_j\Big|_{t=t_o} \tag{2.1.18}$$

$$y_{jf} \equiv y_j\Big|_{t=t_f} \tag{2.1.19}$$

$$\lambda_{jf} \equiv \lambda_j\Big|_{t=t_f} \tag{2.1.20}$$

Using the functional relationships summarized above form the augmented function $J^* = J + \mu_\ell \psi_\ell$ where

$$J(y_{io},y_{if},t_o,t_f) = g(y_{io},y_{if},t_o,t_f) \qquad (2.1.21)$$
$$+\int_{t_o}^{t_f}\Big[-H^o(t,y_{io},y_{if},t_o,t_f)$$
$$+ \lambda_j(t,y_{io},y_{if},t_o,t_f)\ \dot{y}_j(t,y_{io},y_{if},t_o,t_f)\Big]\ dt$$

By requiring the trajectory to satisfy certain necessary conditions regarding path, equations (1.2.10) and (1.2.11), the Problem of Bolza has been reduced to the problem of minimizing J, a function of endpoints, subject to the ψ_ℓ algebraic constraints on the endpoints.

Before proceeding with the minimization of J, it is appropriate to consider a graphical interpretation of the functional relationship for the state variables expressed in equation (2.1.9). Figure 2.1 shows a general state function $y_i(t,y_{io},y_{if},t_o,t_f)$ as a function of time. From the figure it can be seen that a change in the final state Δy_{if} while holding all of the other endpoints fixed causes a change in y_i for all values of t. Likewise a change in the final time Δt_f while holding all of the other endpoints fixed causes a change in the state y_i for all values of t.

Formalizing this graphical interpretation in terms of differentials yields results which will be of value in the following sections. Using equation (2.1.9), the

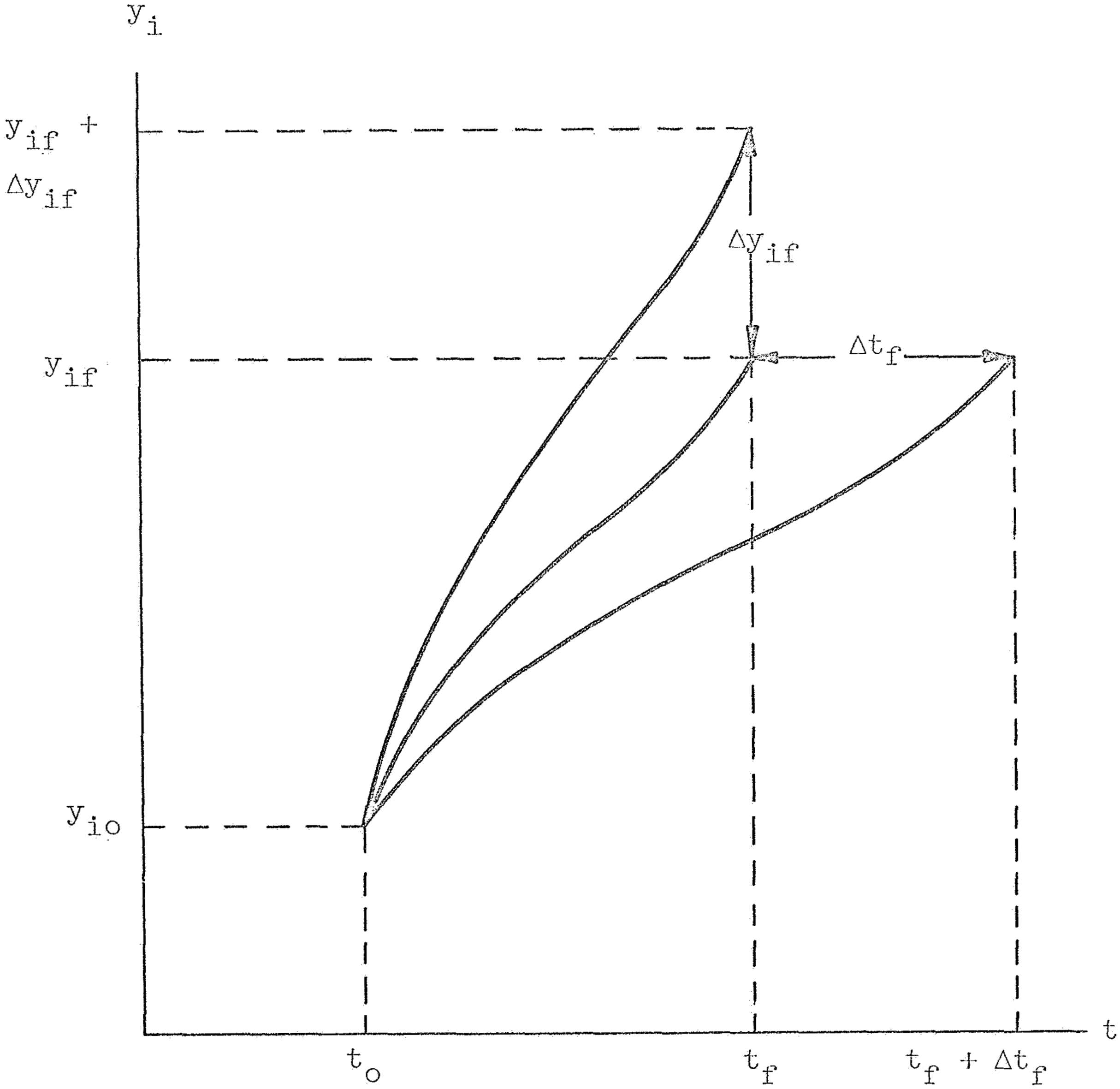

Fig. 2.1 A Representation of y_i as a Function of y_{if} and t_f'

differential of the state variables may be written as,

$$dy_i(t,y_{jo},y_{jf},t_o,t_f) = \frac{\partial y_i}{\partial t}\,dt + \frac{\partial y_i}{\partial t_o}\,dt_o + \frac{\partial y_i}{\partial t_f}\,dt_f \quad (2.1.22)$$
$$+ \frac{\partial y_i}{\partial y_{jo}}\,dy_{jo} + \frac{\partial y_i}{\partial y_{jf}}\,dy_{jf}$$

Evaluating this expression at $t = t_f$ gives

$$dy_i(t_f,y_{jo},y_{jf},t_o,t_f) = dy_{if} = \left.\frac{\partial y_i}{\partial t}\right|_f dt_f \qquad (2.1.23)$$
$$+ \left.\frac{\partial y_i}{\partial t_o}\right|_f dt_o + \left.\frac{\partial y_i}{\partial t_f}\right|_f dt_f + \left.\frac{\partial y_i}{\partial y_{jo}}\right|_f dy_{jo} + \left.\frac{\partial y_i}{\partial y_{jf}}\right|_f dy_{jf}$$

The sum represented by the last term in the above equation can be separated into those products for which $i \neq j$ and that for which $i = j$. Transposing dy_{if} to the right hand side, equation (2.1.23) becomes

$$0 = \left[\left.\frac{\partial y_i}{\partial t}\right|_f + \left.\frac{\partial y_i}{\partial t_f}\right|_f\right] dt_f + \left.\frac{\partial y_i}{\partial y_{jo}}\right|_f dy_{jo}$$
$$+ \left.\frac{\partial y_i}{\partial y_{jf}}\right|_{\substack{f\\ i\neq j}} dy_{if} + \left[\left.\frac{\partial y_i}{\partial y_{jf}}\right|_f - 1\right] dy_{jf}\Big|_{i=j} \qquad (2.1.24)$$

In the above equation the repeated subscripts on the last term do not imply summation. Since t, t_o, t_f, y_{io}, and y_{if} have been assumed to be independent, equation (2.1.24) implies that

$$\left.\frac{\partial y_i}{\partial t}\right|_f = -\left.\frac{\partial y_i}{\partial t_f}\right|_f \tag{2.1.25}$$

$$\left.\frac{\partial y_i}{\partial t_o}\right|_f = 0 \tag{2.1.26}$$

$$\left.\frac{\partial y_i}{\partial y_{jo}}\right|_f = 0 \tag{2.1.27}$$

$$\left.\frac{\partial y_i}{\partial y_{jf}}\right|_{\substack{f\\ i\neq j}} = 0 \tag{2.1.28}$$

$$\left.\frac{\partial y_i}{\partial y_{jf}}\right|_{\substack{f\\ i=j}} = 1 \tag{2.1.29}$$

By evaluating equation (2.1.22) at $t = t_o$ and following arguments similar to the ones above, it can be shown that

$$\left.\frac{\partial y_i}{\partial t}\right|_o = -\left.\frac{\partial y_i}{\partial t_f}\right|_o \tag{2.1.30}$$

$$\left.\frac{\partial y_i}{\partial t_f}\right|_o = 0 \tag{2.1.31}$$

$$\left.\frac{\partial y_i}{\partial y_{jf}}\right|_o = 0 \tag{2.1.32}$$

$$\left.\frac{\partial y_i}{\partial y_{jo}}\right|_{\substack{o \\ i\neq j}} = 0 \qquad (2.1.33)$$

$$\left.\frac{\partial y_i}{\partial y_{jo}}\right|_{\substack{o \\ i=j}} = 1 \qquad (2.1.34)$$

These identities will be useful in the proofs of necessary and sufficient conditions in the next section.

2.2 Derivation of Transversality Conditions

In determining the functional relationships in the last section, it was assumed that the control and adjoint variables were chosen so as to satisfy the Euler-Lagrange Equations (1.2.10) and (1.2.11). Equations (1.2.10) and (1.2.11) are referred to as the first path necessary conditions. In this section the endpoint necessary conditions (transversality conditions) are derived assuming that the first necessary conditions for path are satisfied.

The solution to the path necessary conditions determines one or more trajectories (see section 1.3), any of which may be expressed functionally as a set $[y_j(t,y_{io},y_{if},t_o,t_f),\lambda_j(t,y_{io},y_{if},t_o,t_f),\ u_k(t,y_{io},y_{if},t_o,t_f)]$ as shown in section 2.1. Once the functions representing one of these trajectories is substituted into the integral in equation (2.1.21), the integration can be performed. It is therefore clear that once the trajectory is specified, J is a function of only the parameters y_{io}, y_{if}, t_o and t_f. Specifying the path reduces the problem of minimizing J to the well-known problem of finding the minimum of a function of several variables subject to algebraic equations of constraint (Bryson and Ho, 1969).

It is shown in Appendix A that if the arguments of J in equation (2.1.21) are to satisfy the constraints and minimize J, then it is necessary that the partial derivatives of the auxiliary function, J^* shown below, with respect to y_{io}, y_{if}, t_o, and t_f all be equal to zero.

The J function is defined by $J^* = J + \mu_\ell \psi_\ell$ where J is given by equation (2.1.21). Using the definition of the function G from equation (1.2.8), J^* may be functionally represented as

$$J^*[y_{io},y_{if},t_o,t_f,\mu_\ell] = G\,[y_{io},y_{if},t_o,t_f,\mu_\ell]$$

$$+\int_{t_o}^{t_f} \left[-H^o(y_i,\lambda_i,u_k{}^o,t) + \lambda_i \frac{\partial y_i}{\partial t} \right] \; dt \qquad (2.2.1)$$

In the above equation it is understood that y_i, λ_i, and u_k are all functions of the set $(t,y_{io},y_{if},t_o,t_f)$. In writing the functional relationship shown above, it has been assumed that the controls u_k have been chosen in an optimal fashion in accordance with the control variable

Euler-Lagrange Equation (1.2.11). This is indicated by the superscript o on u_k and on H. The partial derivative of J^* with respect to y_{io} can now be written:

$$\frac{\partial J^*}{\partial y_{jo}} = \frac{\partial G}{\partial y_{jo}} + \int_{t_o}^{t_f} \left[- \frac{\partial H}{\partial y_{jo}} + \frac{\partial \lambda_i}{\partial y_{jo}} \frac{\partial y_i}{\partial t} + \lambda_i \frac{\partial^2 y_i}{\partial y_{jo} \partial t} \right] dt \tag{2.2.2}$$

Here Leibnitz Rule (Hildebrand, 1948, p. 360) has been used for differentiation of an integral with respect to a parameter. Using the identity

$$\frac{d}{dt} \left[\lambda_i \frac{\partial y_i}{\partial y_{jo}} \right] = \lambda_i \frac{\partial^2 y_i}{\partial y_{jo} \partial t} + \frac{\partial \lambda_i}{\partial t} \frac{\partial y_i}{\partial y_{jo}} \tag{2.2.3}$$

and expanding $\frac{\partial H}{\partial y_{jo}}$, equation (2.2.2) may be written as

$$\frac{\partial J^*}{\partial y_{jo}} = \frac{\partial G}{\partial y_{jo}} + \int_{t_o}^{t_f} \left[- \frac{\partial H}{\partial y_i} \frac{\partial y_i}{\partial y_{jo}} - \frac{\partial H}{\partial \lambda_i} \frac{\partial \lambda_i}{\partial y_{jo}} \right. \tag{2.2.4}$$

$$\left. - \frac{\partial H}{\partial u_k} \frac{\partial u_k}{\partial y_{jo}} + \frac{\partial \lambda_i}{\partial y_{jo}} \frac{\partial y_i}{\partial t} - \frac{\partial \lambda_i}{\partial t} \frac{\partial y_i}{\partial y_{jo}} \right] dt + \left[\lambda_i \frac{\partial y_i}{\partial y_{jo}} \right]_{t_o}^{t_f}$$

Terms under the integral sign may be combined to give

$$\frac{\partial J^*}{\partial y_{jo}} = \frac{\partial G}{\partial y_{jo}} + \lambda_i \Big|_{t_f} \frac{\partial y_i}{\partial y_{jo}} \Big|_{t_f} - \lambda_i \Big|_{t_o} \frac{\partial y_i}{\partial y_{jo}} \Big|_{t_o} \tag{2.2.5}$$

$$+ \int_{t_o}^{t_f} \left[- \left(\frac{\partial H}{\partial \lambda_i} - \frac{\partial y_i}{\partial t} \right) \frac{\partial \lambda_i}{\partial y_{jo}} - \left(\frac{\partial H}{\partial y_i} + \frac{\partial \lambda_i}{\partial t} \right) \frac{\partial y_i}{\partial y_{jo}} \right.$$

$$\left. + \frac{\partial H}{\partial u_k} \frac{\partial u_k}{\partial y_{jo}} \right] dt$$

Note from equations (2.1.9) and (2.1.10) that once the optimal endpoints have been selected,

$$\frac{\partial y_i}{\partial t} = \frac{dy_i}{dt} \quad \text{and} \quad \frac{\partial \lambda_i}{\partial t} = \frac{d\lambda_i}{dt}.$$

The integral term vanishes, since equations (1.2.3), (1.2.10) and (1.2.11) were used to generate the functional relations (2.1.9) and (2.1.10).

Using equations (2.1.27), (2.1.33) and (2.1.34), it can be concluded that the sums represented by the two remaining terms not containing G in equation (2.2.5) reduce to a single term, $- \lambda_j \Big|_{t_o}$. With these considerations, equation (2.2.5) reduces to

$$\frac{\partial J^*}{\partial y_{jo}} = \frac{\partial G}{\partial y_{jo}} - \lambda_j \Big|_{t_o} = 0 \qquad (2.2.6)$$

By taking the derivative of J^* with respect to y_{jf} and using arguments similar to those just presented (in this case equations (2.1.32), (2.1.28) and (2.1.29) must be taken into account), it can be shown that

$$\frac{\partial J^*}{\partial y_{jf}} = \frac{\partial G}{\partial y_{jf}} + \lambda_j \Big|_{t_f} = 0 \qquad (2.2.7)$$

Two more necessary conditions remain to be derived. These result from taking the partial derivatives of J^* with respect to the remaining two variables,

t_o and t_f. Performing the first of these operations yields

$$\frac{\partial J^*}{\partial t_o} = \frac{\partial G}{\partial t_o} + \int_{t_o}^{t_f} \left[- \frac{\partial H}{\partial t_o} + \frac{\partial \lambda_i}{\partial t_o} \frac{\partial y_i}{\partial t} + \lambda_i \frac{\partial y_i}{\partial t \partial t_o} \right] dt$$

$$- \left[- H + \lambda_i \frac{\partial y_i}{\partial t} \right]_{t_o} \qquad (2.2.8)$$

Here again Leibnitz Rule has been used; this time the limits of integration are functions of the differentiating variable. Using the identity

$$\frac{d}{dt} \left[\lambda_i \frac{\partial y_i}{\partial t_o} \right] = \lambda_i \frac{\partial y_i}{\partial t \partial t_o} + \frac{\partial \lambda_i}{\partial t} \frac{\partial y_i}{\partial t_o} \qquad (2.2.9)$$

and expanding $\frac{\partial H}{\partial t_o}$, equation (2.2.8) may be written as

$$\frac{\partial J^*}{\partial t_o} = \frac{\partial G}{\partial t_o} + \int_{t_o}^{t_f} \left[- \frac{\partial H}{\partial y_i} \frac{\partial y_i}{\partial t_o} - \frac{\partial H}{\partial \lambda_i} \frac{\partial \lambda_i}{\partial t_o} \right.$$

$$\left. - \frac{\partial H}{\partial u_k} \frac{\partial u_k}{\partial t_o} + \frac{\partial \lambda_i}{\partial t_o} \frac{\partial y_i}{\partial t} - \frac{\partial \lambda_i}{\partial t} \frac{\partial y_i}{\partial t_o} \right] dt \qquad (2.2.10)$$

$$+ \left[\lambda_i \frac{\partial y_i}{\partial t_o} \right]_{t_o}^{t_f} - \left[- H + \lambda_i \frac{\partial y_i}{\partial t} \right] \Bigg|_{t_o}$$

Terms outside the integral may be evaluated at the endpoints indicated and terms under the integral sign combined to give

$$\frac{\partial J^*}{\partial t_o} = \frac{\partial G}{\partial t_o} + H\Big|_{t_o} - \lambda_i\Big|_{t_o} \frac{\partial y_i}{\partial t}\Bigg|_{t_o}$$

$$+ \lambda_i\Big|_{t_f} \frac{\partial y_i}{\partial t_o}\Bigg|_{t_f} - \lambda_i\Big|_{t_o} \frac{\partial y_i}{\partial t_o}\Bigg|_{t_o} \qquad (2.2.11)$$

$$+ \int_{t_o}^{t_f} \left[- \left(\frac{\partial H}{\partial y_i} + \frac{\partial \lambda_i}{\partial t}\right)\frac{\partial y_i}{\partial t_o} \right.$$

$$\left. - \left(\frac{\partial H}{\partial \lambda_i} - \frac{\partial y_i}{\partial t} \right) \frac{\partial \lambda_i}{\partial t_o} - \left(\frac{\partial H}{\partial u_k}\right) \frac{\partial u_k}{\partial t_o} \right] dt$$

The integral again vanishes identically for optimal paths. Using equations (2.1.26) and (2.1.30), the three terms outside the integral representing summations can also be equated to zero. With these observations, equation (2.2.11) reduces to

$$\frac{\partial J^*}{\partial t_o} = \frac{\partial G}{\partial t_o} + H\Big|_{t_o} = 0 \qquad (2.2.12)$$

By taking the derivative of J^* with respect to t_f, following a line of reasoning similar to that just given, and using equations (2.1.31) and (2.1.25), it can be shown that

$$\frac{\partial J^*}{\partial t_f} = \frac{\partial G}{\partial t_f} - H\Big|_{t_f} = 0 \qquad (2.2.13)$$

These results are summarized in the following statement:

2.2.1 Transversality Necessary Condition for Endpoints. If a trajectory satisfies the Euler-Lagrange and state variable differential equations, equations (1.2.10), (1.2.11), and (1.2.3), and if the set $E = [y_{io}, y_{if}, t_o, t_f, \mu_\ell]$ satisfies endpoint equations of constraint (1.2.4) and provides a local minimum of J with respect to small allowable variations in the endpoints, then the set E must satisfy equations (2.2.6), (2.2.7), (2.2.12) and (2.2.13).

These latter equations are referred to as the endpoint necessary conditions or, classically, as the transversality necessary conditions.

2.3 Derivation of Endpoint Sufficiency Conditions

In the last section the function J was shown to be a function of the endpoint variables y_{io}, y_{if}, t_o, and t_f when evaluated along an optimal path. The function J is constrained, however, through the p equations of constraint ψ_ℓ of equation (1.2.4). Sufficiency conditions for determining the minimum of a function whose arguments must satisfy algebraic equations of constraint are well known (Vincent, 1969), and, for reference, the sufficiency conditions are derived in matrix notation in Appendix A.

Before presenting a statement of the sufficiency condition, a brief discussion and definition of notation are in order. Since the algebraic equations of constraint for the Problem of Bolza define relationships among the endpoint variables, the endpoint variables are not all independent. Since there are p equations of constraint and (2n + 2) endpoint variables, there are only (2n - p + 2) independent endpoint variables. The p dependent variables are determined by the p equations of constraint. Any p of the variables can be considered to be the dependent variables. The choice is one of convenience. Let the p dependent variables be denoted by the column vector $\underline{w}$ and the remaining (2n - p + 2) independent variables be denoted by the column vector $\underline{v}$. Let the vector $\underline{\psi}$ represent a vector whose elements are the ψ_ℓ constraint functions. Equation (2.3.1) summarizes these relations.

$$\underline{\psi} = \begin{vmatrix} \psi_1(\underline{w},\underline{v}) \\ \psi_2(\underline{w},\underline{v}) \\ \vdots \\ \psi_p(\underline{w},\underline{v}) \end{vmatrix} \qquad \underline{w} = \begin{vmatrix} w_1 \\ w_2 \\ \vdots \\ w_p \end{vmatrix} \qquad v = \begin{vmatrix} v_1 \\ v_2 \\ \vdots \\ v_q \end{vmatrix} \qquad (2.3.1)$$

$$q = 2n - p + 2$$

The identification of the elements of $\underline{v}$ and the elements of $\underline{w}$ with the endpoints y_{io}, y_{if}, t_o, and t_f is arbitrary except that the set of ψ_ℓ equations must contain every element of $\underline{w}$ and, in addition, every ψ_ℓ equation must contain at least one element of $\underline{w}$. It is convenient to define an additional column vector $\underline{r}$, whose first elements are the dependent variables and last elements the independent variables:

$$\underline{r} = \begin{vmatrix} w_1 \\ \vdots \\ w_p \\ v_1 \\ \vdots \\ v_q \end{vmatrix} \qquad (2.3.2)$$

With these vectors define $\left[\frac{\partial J^*}{\partial \underline{r} \partial \underline{r}}\right]$ as a (2n + 2) by (2n + 2) matrix with elements $a_{ij} = \frac{\partial J^*}{\partial r_i \partial r_j}$. Let the matrix Φ be defined by

$$\Phi = - \left[\frac{\partial \psi}{\partial \underline{w}}\right]^{-1} \left[\frac{\partial \psi}{\partial \underline{v}}\right] \qquad (2.3.3)$$

where $\left[\frac{\partial \psi}{\partial \underline{w}}\right]$ is a p by p matrix with elements $a_{ij} = \frac{\partial \psi_i}{\partial w_j}$ and $\left[\frac{\partial \psi}{\partial \underline{v}}\right]$ is a p by q matrix with elements $a_{ij} = \frac{\partial \psi_i}{\partial v_j}$.

It is shown in Appendix A that the Φ matrix is the linear transformation which transforms differential changes in the independent variables into differential changes in the dependent variables. The Φ matrix has p rows and q columns. Finally, define the (2n + 2) by q partitioned matrix Ω as

$$\Omega = \begin{bmatrix} \Phi \\ \hline I \end{bmatrix} \tag{2.3.4}$$

where I represents a q by q identity matrix.

With these definitions the endpoint sufficiency condition may now be stated:

2.3.1 Endpoint Sufficiency Condition. If E represents a set of endpoints and multipliers $[y_{io}, y_{if}, t_o, t_f, \mu_\ell]$ which satisfy the transversality necessary condition for endpoints, then a sufficient condition for the set E to represent a local minimum of the function J with respect to small allowable variations in the endpoints is that the quadratic form

$$d\underline{v}^T \Omega^T \left[\frac{\partial J^*}{\partial \underline{r} \partial \underline{r}} \right] \Omega \; d\underline{v} \tag{2.3.5}$$

in the differentials $d\underline{v}$ must be positive definite when evaluated at the stationary point E.

To implement this sufficiency test, it is necessary to evaluate the elements of the matrix Φ and the elements of the matrix $\left[\frac{\partial J^*}{\partial \underline{r} \partial \underline{r}} \right]$. Evaluation of elements of Φ represents no problem since the functional form of the constraints is specified in the problem statement. However, the analytic evaluation of the second partial derivatives of J^* with respect to the endpoints is not so simple.

The second partial derivatives of J^* can be obtained by taking the partial derivatives of the transversality necessary conditions with respect to the endpoints r_i.

$$\frac{\partial}{\partial r_i}\left(\frac{\partial J^*}{\partial y_{jo}}\right) = \frac{\partial G}{\partial r_i \partial y_{jo}} - \frac{\partial}{\partial r_i}\left(\lambda_j\Big|_{t_o}\right) \qquad (2.3.6)$$

$$\frac{\partial}{\partial r_i}\left(\frac{\partial J^*}{\partial y_{jf}}\right) = \frac{\partial G}{\partial r_i \partial y_{jf}} + \frac{\partial}{\partial r_i}\left(\lambda_j\Big|_{t_f}\right) \qquad (2.3.7)$$

$$\frac{\partial}{\partial r_i}\left(\frac{\partial J^*}{\partial t_o}\right) = \frac{\partial G}{\partial r_i \partial t_o} + \frac{\partial}{\partial r_i}\left(H\Big|_{t_o}\right) \qquad (2.3.8)$$

$$\frac{\partial}{\partial r_i}\left(\frac{\partial J^*}{\partial t_f}\right) = \frac{\partial G}{\partial r_i \partial t_f} - \frac{\partial}{\partial r_i}\left(H\Big|_{t_f}\right) \qquad (2.3.9)$$

where in the above equations $i = 1,2,\ldots, 2n+2$. The functional form of G as a function of the endpoints is specified by the statement of the problem. However, the functions λ_j and H are not known functions of the endpoints until the state variable and Euler-Lagrange differential equations have been integrated analytically.

Since analytical integration is often difficult or impossible, it would be desirable to evaluate the partial derivatives of λ_i and H with respect to the end-points in terms of functional forms specified in the statement of the problem. A complete set of relationships of this type were not found. Unless future investigators establish such relationships, analytic application of the sufficiency condition requires an analytic solution of the state variable and Euler-Lagrange differential equations.

Some interesting relations of this type are easily obtained however. Each of the elements of the matrix $\left[\frac{\partial J^*}{\partial \underline{r} \partial \underline{r}}\right]$ is composed of a sum of a second partial derivative of G and a second term. The matrix can therefore be expressed as the sum of two matrices,

$$\left[\frac{\partial J^*}{\partial \underline{r} \partial \underline{r}}\right] = \left[\frac{\partial G}{\partial \underline{r} \partial \underline{r}}\right] + A \qquad (2.3.10)$$

where the matrix A is determined from equations (2.3.6) - (2.3.9). Since J^* and G are of class C^2 by hypothesis, both J^* and G must be symmetric about their major diagonals. The obvious conclusion is that matrix A must also be symmetric. By equating symmetric elements of A, the following identities can be established:

$$\frac{\partial \lambda_{io}}{\partial y_{jo}} = \frac{\partial \lambda_{jo}}{\partial y_{io}} \qquad \frac{\partial \lambda_{if}}{\partial y_{jf}} = \frac{\partial \lambda_{jf}}{\partial y_{if}} \qquad (2.3.11)$$

$$\frac{\partial H_o}{\partial y_{io}} = -\frac{\partial \lambda_{io}}{\partial t_o} \qquad \frac{\partial H_f}{\partial y_{if}} = -\frac{\partial \lambda_{if}}{\partial t_f} \qquad (2.3.12)$$

$$\frac{\partial H_f}{\partial y_{io}} = \frac{\partial \lambda_{io}}{\partial t_f} \qquad \frac{\partial H_o}{\partial y_{if}} = \frac{\partial \lambda_{if}}{\partial t_o} \qquad (2.3.13)$$

$$\frac{\partial \lambda_{if}}{\partial y_{jo}} = -\frac{\partial \lambda_{jo}}{\partial y_{if}} \qquad \frac{\partial H_f}{\partial t_o} = -\frac{\partial H_o}{\partial t_f} \qquad (2.3.14)$$

In addition, the following relations can be established by considering the functional relationships exhibited in section 2.1.

$$\frac{\partial H_f}{\partial t_f} = \frac{\partial H}{\partial t} + f_{if}\frac{\partial \lambda_{if}}{\partial t_f} \qquad (2.3.15)$$

$$\frac{\partial H_f}{\partial y_{if}} = \frac{\partial \lambda_{if}}{\partial t_f} + f_{jf}\frac{\partial \lambda_{jf}}{\partial y_{if}} \qquad (2.3.16)$$

Similar equations exist for the initial point.

Unfortunately, a sufficient number of these relationships have not been found to determine the elements of A in terms of known functions in the problem statement. The determination of further relationships and the ultimate determination of the elements of A without resort to analytical integration of the state variable and Euler-Lagrange equations poses an interesting problem for future investigations.

2.4 Relations to Classical Theory

Bolza (1961, pp. 102-103) gives an excellent summary of the various classical approaches to the development of necessary and sufficient conditions for variable endpoint problems. The classical problem in the calculus of variations is to minimize the integral of a function F = F(x,y,y') unconstrained by differential equations of constraint. Here x is the independent variable, y is the independent variable, and y' represents dy/dx. The first and second order variation of the integral are written as δJ and $\delta^2 J$, respectively, while δy and δx represent variations in the dependent and independent variables.

Because of the pertinence of Bolza's remarks to this presentation, his historical synopsis is quoted in detail:

> Three essentially different methods have been proposed for the discussion of problems with variable end-points:
>
> 1. The method of the Calculus of Variations proper: It consists in computing δJ and $\delta^2 J$ either by means of Taylor's formula or by the method of differentiation with respect to ε, . . . and discussing the conditions $\delta J = 0$, $\delta^2 J \geq 0$. The method was first used by LAGRANGE . . . [(1867, pp. 338, 345)]. He gives the general expression for δJ when the endpoints are variable, viz.:

$$\delta J = \int_{x_0}^{x_f} \delta y \left(F_y - \frac{d}{dx} F_{y'} \right) dx + [F\delta x + F_{y'}\delta y]_o^f \tag{2.4.1}$$

and derives the conditions arising from $\delta J = 0$.

The second variation for the case of variable end-points was first developed by Erdmann . . . [(1878, p. 364)]. He finds

$$\delta^2 J = \int_{x_0}^{x_f} \frac{R(u\delta y' - u'\delta y)^2 dx}{u^2} \tag{2.4.2}$$

$$+ \Bigg[F\delta^2 x + F_{y'}\delta^2 y + 2F_{y'}\delta x\delta y + 2F_{y'}\delta x\delta y' + \frac{dF}{dx}\delta x^2 + \left(F_{y'y} + F_{y'y'}\frac{u'}{u} \right) \delta y^2 \Bigg]_0^f$$

where u is an integral of Jacobi's differential equation . . . [(Bolza, 1961, p. 49)]. By considering such special variations for which $\delta y = Cu$, he makes the integral vanish and thus reduces the question to the discussion of the sign of the remaining function of the variations δx_i, δy_i, $\delta^2 x_i$, $\delta^2 y_i$. These variations are connected by relations which depend upon the special nature of the initial conditions . . . .

For the general integral

$$J = \int_{x_0}^{x_f} F(x,y_1,y_2,\ldots,y_n;y_1',y_2',\ldots,y_n')dx \tag{2.4.3}$$

where $y_1,y_2,\ldots,y_n$ are connected by a number of finite or differential relations, the second variation in the case of variable endpoints was studied by A. Mayer . . . [(1896, p. 436)]; for the integral in parameter-representation

$$J = \int_{t_o}^{t_f} F(x,y,x',y')dt \qquad (2.4.4)$$

> by Bliss . . . [(1902, p. 132)].
>
> 2. The method of Differential Calculus: This method is explained in a general way by Dienger . . . [(1867)]. It decomposes the problem into two problems by first considering variations which leave the end-points fixed, and then variations which vary the end-points, the neighboring curves considered being themselves extremals. The second part of the problem reduces to a problem of the theory of ordinary maxima and minima. This method has been used by A. Mayer in an earlier paper on the second variation in the case of variable end-points for the general type of integrals mentioned above . . . [(Mayer, 1884, p. 99)]. It is superior to the first method not only on account of its greater simplicity and its more elementary character, but because--by utilizing the well-known sufficient conditions for ordinary maxima and minima--it leads, in a certain sense, to sufficient conditions if combined with Weierstrass's sufficient conditions for the case of fixed end-points . . . .
>
> 3. Kneser's method: This method, which has been developed by Kneser . . . [(1900)], is based upon an extension of certain well-known theorems on geodesics. It leads in the simplest way to sufficient conditions, but must be supplemented by one of the two preceding methods for an exhaustive treatment of the necessary conditions . . . .

Later investigators of variable endpoint problems, e.g., (Householder, 1937), (Bliss, 1946), and (Hestenes, 1966) have followed the first method quoted from Bolza, "the method of the Calculus of Variations proper." To the best knowledge of the author, there have been no further developments or exposés using "the method of Differential Calculus" since that of Bolza (1904).

The method developed in this presentation is essentially the second method, "the method of Differential Calculus," quoted above from Bolza. Paticular attention should be paid to his remarks concerning this method. These remarks are consistent with fundamental propositions of the last three sections, namely:

> A comprehensive sufficiency condition for variational problems with variable endpoints is obtained by applying two independent tests, Test A and Test B below, each of which is applied separately.
>
> Test A. Satisfaction of the endpoint sufficiency condition given that the trajectory satisfies necessary path conditions.
>
> Test B. Satisfaction of the path sufficiency conditions for the endpoint fixed.

In addition, Bolza's remarks indicate that

> For a solution to be optimal, it is necessary and sufficient that its endpoints satisfy the endpoint sufficiency condition (Test A above) and that its path satisfy the fourth necessary condition with its endpoints considered fixed (Test B above).

Although this presentation has not undertaken a rigorous proof os this hypothesis, the examples and analysis have given every indication that the hypothesis is valid.

2.5 Geodetic Example

As an example of the application of the sufficiency condition for endpoints, consider the problem of determining the minimum distance from the origin to any point on a parabola of the form

$$y = x^2 + b \qquad (2.5.1)$$

In control notation the problem may be formulated as follows:

Minimize

$$J = \int_{s_o}^{s_f} ds \qquad (2.5.2)$$

subject to the state variable differential constraints,

$$\frac{dx}{ds} = \cos g, \qquad (2.5.3)$$

$$\frac{dy}{ds} = \sin g, \qquad (2.5.4)$$

and endpoint constraints,

$$y_o = 0, \qquad (2.5.5)$$

$$x_o = 0, \qquad (2.5.6)$$

$$s_o = 0, \qquad (2.5.7)$$

$$y_f - x_f^2 - b = 0. \qquad (2.5.8)$$

The angle g is the angle between the positive x axis and a tangent to the curve. Here x and y are the state variables, g is the control variable, and s is the independent variable analogous to t in the formulation of earlier sections.

2.5.1 Necessary Path Conditions. The H and G functions are

$$H = \lambda_x \cos g + \lambda_y \sin g - 1 \qquad (2.5.9)$$

$$G = \mu_1(y_f - x_f^2 - b) + \mu_2 y_o + \mu_3 x_o + \mu_4 s_o \tag{2.5.10}$$

The adjoint-variable Euler-Lagrange equations are

$$\dot{\lambda}_y = 0 \tag{2.5.11}$$

$$\dot{\lambda}_x = 0 \tag{2.5.12}$$

and the control-variable Euler-Lagrange equation is

$$-\lambda_x \sin g + \lambda_y \cos g = 0, \tag{2.5.13}$$

Equations (2.5.11) and (2.5.12) imply that λ_x and λ_y are constants. Solving equation (2.5.13) for the control

$$\tan g = \frac{\lambda_y}{\lambda_x} = \text{constant} \tag{2.5.14}$$

which implies

$$\sin g = \frac{\lambda_y}{\sqrt{\lambda_x^2 + \lambda_y^2}} \tag{2.5.15}$$

$$\cos g = \frac{\lambda_x}{\sqrt{\lambda_x^2 + \lambda_y^2}} \tag{2.5.16}$$

The positive sign on the radical is a consequence of the Legendre-Clebsh necessary condition (1.2.17).

2.5.2 Functional Relations. Integrating the state variable equations (2.5.3) and (2.5.4) with the optimal constant control g between the general initial point (x_o, y_o, s_o) and general final point (x_f, y_f, s_f) results in

$$x_f - x_o = (s_f - s_o) \cos g \tag{2.5.17}$$

$$y_f - y_o = (s_f - s_o) \sin g \tag{2.5.18}$$

Solving for the control

$$\tan g = \frac{y_f - y_o}{x_f - x_o} \tag{2.5.19}$$

Squaring both sides of equations (2.5.17) and (2.5.18) and adding yields the identity

$$(s_f - s_o)^2 = (x_f - x_o)^2 + (y_f - y_o)^2 \tag{2.5.20}$$

Solving equations (2.5.17) and (2.5.18) for the controls gives

$$\cos g = \frac{x_f - x_o}{s_f - s_o} \tag{2.5.21}$$

and

$$\sin g = \frac{y_f - y_o}{s_f - s_o} \tag{2.5.22}$$

Since the control is constant, the control is not a function of the independent variable in this case. For other problems the control may be a function of the independent variable as well as the endpoints.

Integrating the state variable equations again between the general initial point (x_o, y_o, s_o) and general intermediate point (x, y, s) and substituting the optimal control from equations (2.5.21) and (2.5.22), and rearranging yields

$$x = x_o + \frac{x_f - x_o}{s_f - s_o}(s - s_o) \tag{2.5.23}$$

$$y = y_o + \frac{y_f - y_o}{s_f - s_o}(s - s_o) \tag{2.5.24}$$

It is seen from the above equations that the state variables are clearly functions of coordinates of the initial and final state variables and of the initial and final values of the dependent variable.

The first integral of the Euler-Lagrange equations is

$$\lambda_x \cos g + \lambda_y \sin g - 1 = 0 \tag{2.5.25}$$

Solving this equation with equation (2.5.13) for λ_x and λ_y and observing equations (2.5.21) and (2.5.22) gives

$$\lambda_x = \frac{x_f - x_o}{s_f - s_o} = \frac{x_f - x_o}{\sqrt{(x_f-x_o)^2 + (y_f-y_o)^2}} \tag{2.5.26}$$

$$\lambda_y = \frac{y_f - y_o}{s_f - s_o} = \frac{y_f - y_o}{\sqrt{(x_f-x_o)^2 + (y_f-y_o)^2}} \tag{2.5.27}$$

Two forms are given above for the Lagrange multipliers as functions of endpoints; either is correct. If the second set is used, the J^* function will be independent of s_f and s_o. In either case it is clear that the Lagrange multipliers can be written as explicit functions of the coordinates of the initial and final states and of the initial and final values of the dependent variables. Equations (2.5.19), (2.5.23), (2.5.24), (2.5.26), and (2.5.27) bear out the functional dependencies hypothesized for control, state, and adjoint variables in section 2.1.

Note that in deriving these equations, only path necessary conditions have been used. The transversality necessary conditions for endpoints have not been used.

2.5.3 Necessary Endpoint Conditions. The transversality conditions, (2.2.6), (2.2.7), (2.1.12), and (2.2.13) yield the following equations

$$\frac{\partial J^*}{\partial y_o} = \mu_2 - \lambda_{yo} = 0 \qquad (2.5.28)$$

$$\frac{\partial J^*}{\partial x_o} = \mu_3 - \lambda_{xo} = 0 \qquad (2.5.29)$$

$$\frac{\partial J^*}{\partial s_o} = \mu_4 + H_o = 0 \qquad (2.5.30)$$

$$\frac{\partial J^*}{\partial y_f} = \mu_1 + \lambda_{yf} = 0 \qquad (2.5.31)$$

$$\frac{\partial J^*}{\partial x_f} = - 2\mu_1 x_f + \lambda_{xf} = 0 \qquad (2.5.32)$$

$$\frac{\partial J^*}{\partial s_f} = - \lambda_{xf} \cos g_f - \lambda_{yf} \sin g_f + 1 = 0 \qquad (2.5.33)$$

Since the initial point is fixed, the initial point transversality equations give no useful information.

To find the optimal endpoints, eliminate μ_1 between equations (2.5.31) and (2.5.32), yielding

$$\lambda_{xf} + 2\lambda_{yf} x_f = 0 \qquad (2.5.34)$$

Substituting λ_{xf} and λ_{yf} from equations (2.5.26) and (2.5.27) into equation (2.5.34) yields

$$\frac{x_f - x_o}{\sqrt{(x_f-x_o)^2 + (y_f-y_o)^2}} + \frac{2(y_f-y_o)\ x_f}{\sqrt{(x_f-x_o)^2 + (y_f-y_o)^2}} = 0 \qquad (2.5.35)$$

Finally, multiplying through by the radical and imposing endpoint constraints (2.5.5) and (2.5.6) gives

$$x_f(1 + 2\ y_f) = 0 \qquad (2.5.36)$$

The necessary conditions are satisfied if either term in the above equation is equal to zero. Solving equation (2.5.36) and equation (2.5.8) simultaneously gives the two solutions

$$\left.\begin{array}{lll} x_f = \pm\sqrt{-\frac{1}{2} - b} & y_f = -\frac{1}{2} & \text{(solution A)} \\ \text{and} & & \\ x_f = 0 & y_f = b & \text{(solution B)} \end{array}\right\} \quad (2.5.37)$$

These endpoints and the corresponding multiple solutions for $b < -\frac{1}{2}$ are shown in Figure 1.3 on page 14. From the symmetry of the parabola, it is expected that either the plus or the minus sign in equation (2.5.37) will determine a solution giving the same value of distance. For this reason a distinction has not been made between the two. The necessary conditions used so far have provided no means for determining under what circumstances solution A (or solution B) is the optimum. In this case of multiple stationary solutions, the endpoint sufficiency condition will provide a means for determining the true optimum.

Before examining the sufficiency conditions, the parameter μ_1 will be evaluated in terms of the general endpoints for future reference. From equations (2.5.27) and (2.5.31) it is observed that

$$\mu_1 = - \lambda_{yf} = - \frac{y_f - y_o}{\sqrt{(x_f - x_o)^2 + (y_f - y_o)^2}} \qquad (2.5.38)$$

2.5.4 Sufficiency Endpoint Condition. To evaluate the endpoint sufficiency conditions, it is instructive to first determine the Φ and Ω matrices of equations (2.3.3) and (2.3.4). The constraints are

$$\psi_1: \qquad y_o = 0 \qquad (2.5.39)$$

$$\psi_2: \qquad x_o = 0 \qquad (2.5.40)$$

$$\psi_3: \qquad s_o = 0 \qquad (2.5.41)$$

$$\psi_4: \qquad y_f - x_f^2 - b = 0 \qquad (2.5.42)$$

Since there are four equations and six endpoints, there are two degrees of freedom. For conveniency let x_f and s_f be the independent variables and y_o, x_o, s_o, and y_f be the dependent variables. Then in the notation of section 2.3

$$\underline{v} = \begin{vmatrix} x_f \\ s_f \end{vmatrix} , \quad \underline{w} = \begin{vmatrix} y_o \\ x_o \\ s_o \\ y_f \end{vmatrix} , \quad \underline{r} = \begin{vmatrix} y_o \\ x_o \\ s_o \\ y_f \\ x_f \\ s_f \end{vmatrix} , \quad \underline{\psi} = \begin{vmatrix} y_o \\ x_o \\ s_o \\ y_f - x_f^2 - b \end{vmatrix} \qquad (2.5.43)$$

Evaluating the matrix of partial derivatives of $\underline{\psi}$ with respect to the independent variables gives

$$\left[\frac{\partial \underline{\psi}}{\partial \underline{v}} \right] = \begin{vmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -2x_f & 0 \end{vmatrix} \qquad (2.5.44)$$

Evaluating the matrix of partial derivatives of $\underline{\psi}$ with respect to the dependent variables gives just the identity matrix

$$\frac{\partial \underline{\psi}}{\partial \underline{w}} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \qquad (2.5.45)$$

The inverse of this matrix is obviously the identity matrix. From equations (2.5.44) and (2.5.45) the Φ matrix can be computed

$$\Phi = - \left[\frac{\partial \psi}{\partial \underline{w}} \right]^{-1} \left[\frac{\partial \psi}{\partial \underline{v}} \right] = \begin{vmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 2x_f & 0 \end{vmatrix} \qquad (2.5.46)$$

The Ω matrix is formed by adjoining to Φ an identity matrix with the dimensions equal to the number of independent variables. In this case there are two independent variables. The Ω matrix is

$$\Omega = \left[\frac{\Phi}{I} \right] = \begin{vmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 2x_f & 0 \\ 1 & 0 \\ 0 & 1 \end{vmatrix} \qquad (2.5.47)$$

With the $\left[\frac{\partial J^*}{\partial \underline{r} \partial \underline{r}} \right]$ not yet evaluated, the endpoint sufficiency condition reduces to the condition that

$$[dx_f \;\; ds_f] \begin{vmatrix} 4x_f^2 \frac{\partial^2 J^*}{\partial y_f^2} + 4x_f \frac{\partial^2 J^*}{\partial y_f \partial x_f} + \frac{\partial^2 J^*}{\partial x_f^2} & 2x_f \frac{\partial^2 J^*}{\partial y_f \partial s_f} + \frac{\partial^2 J^*}{\partial x_f \partial s_f} \\ 2x_f \frac{\partial^2 J^*}{\partial s_f \partial y_f} + \frac{\partial^2 J^*}{\partial s_f \partial x_f} & \frac{\partial^2 J^*}{\partial s_f^2} \end{vmatrix} \begin{vmatrix} dx_f \\ ds_f \end{vmatrix} \qquad (2.5.48)$$

must be positive definite.

If J* can be written so that it is not a function of s_f, the sufficiency condition will be reduced to a simple inequality involving dx_f only. From the transversality equations (2.5.31) - (2.5.33) and the functional relations for λ_x and λ_y, equations (2.5.26) and (2.5.27), it is seen that this can be done.

Therefore, the sufficiency condition reduces to the condition that

$$\left[4x_f^2 \frac{\partial^2 J^*}{\partial {y_f}^2} + 4x_f \frac{\partial^2 J^*}{\partial y_f \partial x_f} + \frac{\partial^2 J^*}{\partial {x_f}^2} \right] dx_f^2 > 0 \qquad (2.5.49)$$

Since dx_f^2 is always positive, the expression in parenthesis must be positive in order to satisfy the sufficiency condition:

$$4x_f^2 \frac{\partial^2 J^*}{\partial {y_f}^2} + 4x_f \frac{\partial^2 J^*}{\partial y_f \partial x_f} + \frac{\partial^2 J^*}{\partial {x_f}^2} > 0 \qquad (2.5.50)$$

This result is identical to the result that would have been obtained if the fixed endpoint coordinates, y_o, x_o, and s_o had been excluded from the G function. This situation is similar to the transversality necessary conditions in that the initial points yield no information. From this example and previous experience with endpoint conditions, the following conclusion is drawn: No useful information concerning either necessary or sufficient conditions results from including in G constraints which merely fix a given endpoint coordinate.

Substituting the functional forms for λ_{xf} and λ_{yf} not involving s_f from equations (2.5.26) and (2.5.27) into the first partial derivatives of J^* with respect to y_f and x_f in equations (2.5.31) and (2.5.32) gives

$$\frac{\partial J^*}{\partial y_f} = \mu_1 + \frac{y_f - y_o}{\sqrt{(x_f-x_o)^2 + (y_f-y_o)^2}} \tag{2.5.51}$$

$$\frac{\partial J^*}{\partial x_f} = - 2\mu_1 x_f + \frac{x_f-x_o}{\sqrt{(x_f-x_o)^2 + (y_f-y_o)^2}} \tag{2.5.52}$$

Forming the required partial derivatives results in

$$\frac{\partial^2 J^*}{\partial x_f^{\,2}} = - 2\mu_1 + \frac{(x_f-x_o)^2 + (y_f-y_o)^2 - x_f(x_f-x_o)}{D} \tag{2.5.53}$$

$$\frac{\partial^2 J^*}{\partial y_f \partial x_f} = - \frac{(x_f-x_o)(y_f-y_o)}{D} \tag{2.5.54}$$

$$\frac{\partial^2 J^*}{\partial y_f^{\,2}} = \frac{(x_f-x_o)^2 + (y_f-y_o)^2 - y_f(y_f-y_o)}{D} \tag{2.5.55}$$

$$\frac{\partial^2 J^*}{\partial x_f \partial y_f} = - \frac{(y_f-y_o)(x_f-x_o)}{D} \tag{2.5.56}$$

where

$$D = [(x_f-x_o)^2 + (y_f-y_o)^2]^{\frac{3}{2}} \tag{2.5.57}$$

Comparing equations (2.5.54) and (2.5.56) verifies the symmetry of the $\left[\frac{\partial^2 J^*}{\partial \underline{r} \partial \underline{r}}\right]$ matrix.

Evaluating these derivatives using initial point constraint equations (2.5.5) and (2.5.6) and μ_1 from equation (2.5.38) and substituting them into the endpoint sufficiency condition, equation (2.5.50) gives

$$\frac{4x_f^4}{(x_f^2 + y_f^2)^{3/2}} - 4x_f \left[\frac{x_f y_f}{(x_f^2 + y_f^2)^{3/2}} \right] + 2 \frac{y_f}{\sqrt{x_f^2 + y_f^2}} + \frac{y_f^2}{(x_f^2 + y_f^2)^{3/2}} > 0 \qquad (2.5.58)$$

For solution B ($x_f = 0$, $y_f = b$) this condition reduces to

$$\frac{1}{b} + 2 > 0 \qquad (2.5.59)$$

From the geometry in Figure 1.3, it can be seen that b is negative. The condition therefore requires that

$$0 > b > -\frac{1}{2} \qquad (2.5.60)$$

Then solution B as shown in Figure 1.3 is optimum.

For solution A ($x_f = \pm \sqrt{-\frac{1}{2} - b}$, $y_f = -\frac{1}{2}$), the end sufficiency condition (2.5.58) becomes

$$\frac{4b^2 + 2b + \frac{1}{4}}{(-\frac{1}{4} - b)^{3/2}} - \frac{1}{\sqrt{-\frac{1}{4} - b}} > 0 \qquad (2.5.61)$$

Combining terms yields

$$\frac{4b^2 + 3b + \frac{1}{2}}{(- \frac{1}{4} - b)^{3/2}} > 0 . \qquad (2.5.62)$$

In order for the denominator to be real

$$b < - \frac{1}{4} . \qquad (2.5.63)$$

Under this condition inequality (2.5.62) is satisfied only if

$$b < - \frac{1}{2} . \qquad (2.5.64)$$

Therefore solution A shown in Figure 1.3 is optimum for b less than $- \frac{1}{2}$. The optimal solution is summarized below.

$$x_f = 0, \qquad y_f = b \qquad 0 > b > - \frac{1}{2} \qquad (2.5.65)$$

$$x_f = \pm \sqrt{- \frac{1}{2} - b}, \qquad y_f = - \frac{1}{2} \qquad b < - \frac{1}{2} \quad (2.5.66)$$

This simple example has been analyzed in great detail to emphasize the concepts developed in earlier sections and to reinforce and illustrate the notation. A more complex example of the endpoint sufficiency condition is presented in Chapter 5.

2.6 A Numerical Algorithm

In order to apply the endpoint sufficiency condition, the matrix of second partial derivatives of J^* with respect to its arguments must be determined. From equations (2.3.6) - (2.3.9) it is seen that each of these second partial derivatives is composed of two terms. The first term, in all cases, is a second partial derivative of the function G. This derivative can be computed analytically from information given in the statement of the problem. The second term of each second partial derivative of J^* can be written in one of the following forms:

$$\frac{\partial M\big|_{t=t_f}}{\partial r_{io}}, \tag{2.6.1}$$

$$\frac{\partial M\big|_{t=t_o}}{\partial r_{if}}, \tag{2.6.2}$$

$$\frac{\partial M\big|_{t=t_f}}{\partial r_{if}}, \tag{2.6.3}$$

or

$$\frac{\partial M\big|_{t=t_o}}{\partial r_{io}}, \tag{2.6.4}$$

where M represents any of the quantities H, λ_1, λ_2, ..., λ_n and r represents any of the state variables y_i or the independent variable t.

These derivatives cannot be evaluated analytically without obtaining an analytic solution to the set of state variable and Euler-Lagrange differential equations. For most problems of practical interest in the calculus of variations, the set of nonlinear state variable and Euler-Lagrange differential equations cannot be integrated analytically. Therefore, the implementation of the endpoint sufficiency condition in most cases requires the numerical computation of partial derivatives of the forms expressed in equations (2.6.1) - (2.6.4).

Fortunately, this is not conceptually difficult for most problems in engineering which have separated end constraints. End constraints are separated if none of the endpoint constraints involves both initial values and final values; the constraints always relate initial values to other initial values, or final values to other final values.

The function M evaluated at $t = t_o$ will be indicated by a subscript o:

$$M_o = M_o(y_{io}, y_{if}, t_o, t_f) \qquad (2.6.5)$$

The function M evaluated at $t = t_f$ will be indicated by a subscript f:

$$M_f = M_f(y_{io},y_{if},t_o,t_f).$$

Before the sufficiency condition test is applied, the problem is first solved using the necessary conditions yielding nominal endpoints y_{io}^*, y_{if}^*, t_o^*, and t_f^* and nominal Lagrange multipliers λ_{io}^* and λ_{if}^*. For brevity, let $\underline{r}_o^*$ represent a vector with elements $(y_{1o}^*, y_{2o}^*, \ldots, y_{no}^*, t_o^*)$, and $\underline{r}_f^*$ represent a vector with elements $y_{1f}^*, y_{2f}^*, \ldots, y_{nf}^*, t_f^*)$, and M^* be a function evaluated with the nominal endpoints.

Numerically the derivative (2.6.1) can be approximated as

$$\frac{\partial M_f}{\partial r_{io}} = \frac{M_f(r_{1o}^*, r_{2o}^*, \ldots, r_{io}^* + \Delta, \ldots; \underline{r}_f^*) - M_f^*}{\Delta} \tag{2.6.7}$$

where Δ is a small change in the nominal initial variable r_{io}^*. If the state variable and Euler-Lagrange equations are then numerically integrated forward with the nominal Lagrange multipliers, the final nominal endpoint will not be reached. The n initial Lagrange multipliers must be adjusted in order to obtain the final nominal endpoint again. Since the n initial Lagrange multipliers give only n degrees of freedom, the nominal endpoint can be reached only if M is a function of n or less than n <u>independent</u> final values. This will be true if there is at least one equation of constraint involving the final values. With these new multipliers, the differential equations are integrated forward to the final point $\underline{r}_f^*$. M_f is then evaluated from the resulting final Lagrange multipliers and $\underline{r}_f^*$. With M_f evaluated, the desired partial derivative can be evaluated using equation (2.6.7).

The derivative (2.6.2) can be approximated numerically as

$$\frac{\partial M_o}{\partial r_{if}} = \frac{M_o(\underline{r}_o^*;\ r_{1f}^*,\ r_{2f}^*,\ \ldots,\ r_{if}^* + \Delta,\ \ldots) - M_o^*}{\Delta} \tag{2.6.8}$$

In the above equation, M_o is evaluated by making a small change in r_{if}^*, while leaving all the other values unchanged. A set of final Lagrange multipliers is then determined so that a backward numerical integration in time will yield the nominal initial values $\underline{r}_o^*$. The quantity M_o is evaluated using the resulting initial Lagrange multipliers and $\underline{r}_o^*$. With M_o computed in this manner, the desired partial derivative can be evaluated using equation (2.6.8).

The derivative (2.7.3) can be approximated numerically as

$$\frac{\partial M_f}{\partial r_{if}} = \frac{M_f(\underline{r}_o^*;\ r_{1f}^*,\ r_{2f}^*,\ \ldots,\ r_{if}^* + \Delta,\ \ldots) - M_f^*}{\Delta} \tag{2.6.9}$$

Here, M_f is evaluated by making a small change in the nominal final point coordinate r^*_{if}, while leaving all of the other final coordinates and the initial point $\underline{r}^*_o$ unchanged. A set of initial Lagrange multipliers is then determined so that a forward integration from the nominal initial point will yield the varied final point $(r^*_{1f}, r^*_{2f}, \ldots, r^*_{if} + \Delta, \ldots)$. The forward integration is then performed to the varied final point, and M_f is evaluated using the resulting final Lagrange multipliers and the coordinates of the varied final point.

The final derivative (2.6.4) can be approximated numerically as

$$\frac{\partial M_o}{\partial r_{io}} = \frac{M_o(r^*_{1o}, r^*_{2o}, \ldots, r^*_{io} + \Delta, \ldots; \underline{r}^*_f) - M^*_o}{\Delta} \qquad (2.6.10)$$

Here, M_o is evaluated by making a small change in the nominal initial point r^*_{io}, while leaving all of the other initial coordinates and the final point $\underline{r}^*_f$ unchanged. A set of final Lagrange multipliers is then determined so that a backward integration in time from the nominal final point will yield the varied initial point $(r^*_{1o}, r^*_{2o}, \ldots, r^*_{io} + \Delta, \ldots)$. The backward integration is then performed to the varied initial point, and M_o is evaluated using the resulting initial Lagrange multipliers and the coordinates of the varied initial point.

Using the above techniques, the matrix of second partial derivatives of J^* with respect to its arguments can be evaluated. Because of the identities (2.3.11) - (2.3.14), there is some choice as to which of the above derivatives is used to evaluate the sufficiency condition. It is a simple matter to numerically evaluate the matrix Ω from the nominal initial and final points and to test the matrix $\Omega^T \frac{\partial^2 J^*}{\partial \underline{r} \partial \underline{r}} \Omega$ for positive-definitness. The details of programs written to perform these operations are left for discussion in Chapter 3.

Selecting the correct initial Lagrange multipliers so that the desired final points are reached as a result of integration is termed a problem with mixed end conditions, or a two-point boundary value problem. The numerical implementation of the sufficiency condition depends strongly upon the existence of numerical techniques for solving two point boundary value problems. These techniques are explained in detail in the next chapter.

CHAPTER 3

NUMERICAL SOLUTION OF TWO-POINT BOUNDARY VALUE PROBLEM

Whether one attacks optimal control problems from the point of view of the classical calculus of variations or by application of the Maximum Principle, one invariably must solve a set of ordinary differential equations subject to both initial and final boundary conditions. Such problems are referred to as two-point boundary value problems. The theory of ordinary differential equations subject to initial conditions has been well developed, both analytically and computationally, for some time. However, the theory of two-point boundary value problems has been slower to develop, especially from a computational point of view. In the past decade, however, significant advances have been made in the application of large scale digital computers to the solution of two-point boundary value problems. The primary motivation for these advances has come from the study of optimal control theory (Handelsman, 1966; McGill and Kenneth, 1964).

This chapter discusses the use of a numerical technique for solving two-point boundary value problems called the generalized Newton-Raphson method. Section 3.1 briefly describes the relationship of this method to other numerical algorithms currently available. The method is explained in section 3.2 and a brief discussion of the actual computational procedure is given in section 3.3.

3.1 Numerical Methods Available

Currently, there are many numerical algorithms for solving two-point boundary value problems; the current question is one of deciding which to implement. To aid in this decision a brief comparison of available computational algorithms is now made. The point of view of Kalman (1964) is taken in the discussion which follows.

The computational solution of optimal control problems always requires that the following five conditions be satisfied:

(1) state-variable differential equations (1.2.3)

(2) algebraic equations constraining state variable endpoints (1.2.4)

(3) transversality algebraic equations (1.2.12) - (1.2.15)

(4) optimal control conditions: $\max_u H(y_i,\lambda_i,u_k,t)$ (equation (1.2.11) or (1,6.2))

(5) adjoint variable Euler-Lagrange differential equations (1.2.19)

The conditions above define a two-point boundary value problem. Five computational algorithms which attempt to satisfy these conditions by successive convergent approximations are discussed below.

3.1.1 Flooding Technique. A great deal can be learned about the optimal control of a system without attempting to obtain a solution to the two-point boundary value problem. If there are p unknown initial values, the problem is first relaxed by not requiring p conditions of (2), which constrain a final state coordinate, to be satisfied. The problem then becomes an initial value problem with one or more arbitrary initial values. If the number of arbitrary initial values is one or at most two, the method of flooding (Vincent and Brusch, 1966) is practical.

Solutions are generated satisfying conditions (1), (3), (4), and (5) and initial conditions of (2). The equations of conditions (3) and (4) are combined, eliminating the μ_ℓ parameters, to yield a cutoff function. Arbitrary values for the unspecified initial variables are chosen, and the resulting initial value problem is integrated numerically until one of the zeros of the cutoff function is encountered, indicating the transversality conditions have been met. Using this method, the state variable endpoints corresponding to the arbitrary initial values simply fall where they may. A family of solutions is generated using the above technique by systematically varying the initial parameters throughout their range. The family specifies the region of endpoint space throughout which solutions are possible. In addition, a mapping is obtained between unknown initial values and final endpoints.

Since only a single integration is needed for each solution, this method is economical for parametric studies when compared with more sophisticated techniques described later. The technique economically generates such a large number of results that solutions to all points in endpoint space may be easily visualized and areas of unusual interest quickly discovered. From the resultant manifold of solutions, the solution to a particular two-point boundary value problem can be readily approximated.

The disadvantages of this technique are the following:

(1) Exact solutions to particular endpoints are not obtained. It is desirable, in the case of multiple solutions, to compare directly two differing solutions to the same endpoint.

(2) Often no optimal endpoints exist for certain ranges of initial values of the Lagrange multipliers (Vincent and Brusch, 1966, pp. 27-28). These ranges are difficult to predict and integrations with initial values within these ranges yield no useful information.

3.1.2 The Classical Indirect Method. To use this procedure, a series of solutions satisfying conditions (1), (4) and (5) are generated (see pages 70-71). This series converges to a solution satisfying conditions (2) and (3). The usual method for satisfying conditions (2) and (3) is to numerically generate a matrix of partial derivatives of the final coordinates with respect to unknown initial values. This is done by making a small change in one of the unknown initial values and observing the resulting changes in the final coordinates. By considering the final coordinates to be linear functions of the unknown initial values, a simple matrix inversion yields new initial values and the iteration is repeated.

The method requires good guesses for the initial values of the adjoint variables, particularly for high dimensional state spaces. The variations in the terminal boundary conditions with variations in the unknown initial values are often so sensitive that numerical solutions are impractical. For further discussion, refer to Kopp and Moyer (1966, pp. 105-114).

3.1.3 Method of Gradients. This procedure is initiated by making an initial estimate of the control as a function of time; the corresponding trajectory is computed, and the value of the performance index calculated. The "direction" in state space for which the rate of change of the performance index is greatest is then determined as a function of time. A small step is then taken in the gradient direction. That is, at each point in time, each state space coordinate is modified by a small quantity proportional to the "projection" of the gradient vector on its coordinate axis. The performance index is reevaluated and the iteration proceeds until the value of the index is stationary. As opposed to the indirect methods previously discussed which solve the first order necessary conditions on p. 70, this method is considered a direct method, since a direct search for the extreme value of the performance index is made. Methods previously discussed have been indirect since solutions have been generated using necessary conditions of the Calculus of Variations.

In terms of the necessary conditions, listed on p. 70, conditions (1) and (5) are satisfied by each solution of a series of solutions, and iteration proceeds until (2), (3) and (4) are satisfied within some tolerance. Convergence of this method does not depend upon good initial guesses for the controls. Although the method is relatively easy to program, its efficiency is limited, since the convergence slows in the neighborhood of the optimal solution. For a complete discussion of this theory, consult Kelly (1962, pp. 205-254). A detailed discussion of a numerical algorithm with an example is found in Hillsley and Robbins (1964, pp. 107-134).

3.1.4 The Second Variation Method. The second variation method (Kelly, Kopp, and Moyer, 1964) is also termed a direct method, since a direct search is made on the functional. Again a series of solutions satisfying conditions (1) and (5) is generated and iteration of the solution proceeds until conditions (2), (3), and (4) are satisfied. The main advantage of this method over the method of gradients is that convergence in the neighborhood of the solution is much improved. The second variation method determines the step size to be taken, while the gradient method provides no indication of step size. Again, convergence to a local minimum is independent of the initial solution estimates. Unfortunately, the computer time saved by this method is offset by computer programming which is significantly more complicated than for the gradient method.

3.1.5 The Generalized Newton-Raphson Method. This method, which will be discussed in detail in the next section, is an indirect method. It differs from classical indirect methods only in the technique used to solve the two-point boundary value problem. Necessary conditions (2), (3), and (4) are satisfied by each solution of a series of solutions, and an iteration is performed in function space which converges to a set of functions satisfying conditions (1) and (5). The determination of the step size is inherent in the method, and convergence is rapid near the optimal solution. Programming for this method is more complex than for the gradient method, but less involved than that required for the second variation method.

The convergence of this method depends upon the choice of initial solution estimates. A sufficiency theorem for convergence has been developed by McGill and Kenneth (1963) for a general system of n second order ordinary nonlinear differential equations. Unfortunately, as the authors point out, problems of engineering interest seldom satisfy the stringent conditions of the theorem. The conditions of the theorem are sufficient, but not necessary, and many investigators, including this author, have achieved convergence from elementary initial estimates which do not satisfy the conditions of the theorem (Moyer and Pinkham, 1964, pp. 91-106; Lewallen, 1967). Sophisticated and effective algorithms have been developed by Lewallen, Tapley, and Williams (1968) for stabilizing convergence of the iteration at some sacrifice in efficiency. With these techniques, convergence envelopes have been achieved which permit initial estimates, generated with some of the initial values in error by over 100 per cent, to converge to the optimal solution. If initial estimates are in too great of error, the iteration diverges.

3.2 A Generalized Method of Newton-Raphson

The Newton-Raphson method was first suggested by Hestenes (1949) for obtaining solutions to fixed endpoint problems in the calculus of variations. The method was expanded and developed by Bellman and Kalba (1965) to include problems with a variety of boundary conditions. Bellman and Kalba referred to the method as quasilinearization. The method was applied to n-dimensional optimization problems formulated in state variable, control variable notation by McGill and Kenneth (1964). The following discussion follows their work, except for the section on variable endpoint problems with final time free.

3.2.1 An Iteration for Nonlinear Initial Value Problems.

Consider the following system of 2n nonlinear differential equations:

$$\dot{\underline{y}} = \underline{F}(\underline{y},t) \qquad (3.2.1)$$

subject to fixed initial point constraints

$$y_j(0) = y_{jo} \qquad j = 1,2,\ldots,p \qquad (3.2.2)$$

and final point constraints

$$y_k(t_f) = y_{kf} \qquad k = 1,2,\ldots,2n\text{-}p \qquad (3.2.3)$$

where

$$\underline{F} = (f_1,f_2,\ldots,f_{2n}) \qquad (3.2.4)$$

$$f_i = f_i(y_1,y_2,\ldots,y_{2n},t) \quad i = 1,2,\ldots,2n \qquad (3.2.5)$$

$$\underline{\psi} = (\psi_1,\psi_2,\ldots,\psi_{2n-p}) \qquad (3.2.6)$$

If $\underline{F}$ is a function of t, $\underline{F}$ can be transformed into an autonomous system by observing the change of variable

$$y_{2n+1} = t \qquad (3.2.7)$$

$$\dot{y}_{2n+1} = 1 \qquad (3.2.8)$$

and adding the constraint

$$y_{(2n+1)o} = 0 \qquad (3.2.9)$$

Thus, $\underline{F}$ may be considered as a function of only $\underline{y}$ without any loss in generality; such a functional relationship is assumed in the following development.

Consider the truncated Taylor Series expansion of $\underline{F}(\underline{y})$ about some nominal state function $\underline{y}^o(t)$ which satisfies initial condition constraints but does not necessarily satisfy equation (3.2.1):

$$\underline{F}\,[\underline{y}^1(t)] \cong \underline{F}\,[\underline{y}^o(t)] + \left[\frac{\partial \underline{F}}{\partial \underline{y}}\right]_o [\underline{y}^1(t) - \underline{y}^o(t)] \quad (3.2.10)$$

where

$$\underline{y}(t) = [y_1(t), y_2(t), \ldots, y_{2n}(t)] \quad (3.2.11)$$

and $\left[\frac{\partial \underline{F}}{\partial \underline{y}}\right]$ is recognized as the Jacobian of the vector function. Here $\underline{y}^o(t)$ may be identified as an estimate to the solution of equation (3.2.1). If, at any time t, a new trajectory $\underline{y}^1(t)$ is not equal to the estimate $\underline{y}^o(t)$, equation (3.2.10) gives an approximate formula for estimating values of the vector function $\underline{F}\,[\underline{y}^1(t)]$. By substituting the linear approximation for $\underline{F}(\underline{y}^1)$ from equation (3.2.10) into equation (3.2.1), a linearized approximation of the nonlinear differential equation results:

$$\dot{\underline{y}}^1(t) \cong \left[\frac{\partial \underline{F}}{\partial \underline{y}}\right]_o \underline{y}^1(t) + \left\{ \underline{F}[y^o(t)] - \left[\frac{\partial \underline{F}}{\partial \underline{y}}\right]_o y^o(t) \right\} \quad (3.2.12)$$

Since $\left[\frac{\partial \underline{F}}{\partial \underline{y}}\right]_o$ and $\underline{y}^o(t)$ are known functions of time, equation (3.2.12) is recognized as a vector linear differential equation with time varying coefficients. The quantity in brackets is the driving function, a known function of time.

It is assumed that the solution of this linearized differential equation will give an approximate solution to the nonlinear differential equation subject to a given set of initial conditions. In particular, suppose that the solution $\underline{y}^1(t)$ found by integrating equations (3.2.12) is, in some sense, a better approximation to the solution of equation (3.2.1) than $\underline{y}^o(t)$. If this is the case in general, an even better approximation for the solution of the nonlinear equation could be obtained by discarding $\underline{y}^o(t)$ and integrating the linearized equation again, this time regarding the new solution $\underline{y}^1(t)$ as the solution estimate. Evidently this process could be repeated any number of times. Iteration (n + 1) can be written as

$$\dot{\underline{y}}^{n+1} = \left[\frac{\partial \underline{F}}{\partial \underline{y}}\right]_n \underline{y}^{n+1} + \left[\underline{F}(\underline{y}^n) - \left[\frac{\partial \underline{F}}{\partial \underline{y}}\right]_n \underline{y}^n\right] \qquad (3.2.13)$$

The superscripts here are iteration indices and do not represent exponentiation.

The iteration represented in equation (3.2.13) is termed a generalized Newton-Raphson iteration, since it may be obtained from a direct generalization of the Newton-Raphson operator for finding roots of a scalar equation (McGill and Kenneth, 1964, p. 1762). The question of convergence of this sequence has been purposely avoided, since even heuristic arguments are quite complex. It is again noted that a sufficiency theorem for convergence does exist for systems which satisfy certain stringent requirements on the set of differential equations and on $\underline{y}^{o}(t)$ (McGill and Kenneth, 1963).

3.2.2 Problems with Fixed Boundary Coordinates. Up until this point the boundary conditions of equation (3.2.3) have been ignored. The above arguments have simply shown that a series of solutions to a linear initial value problem will, under appropriate circumstances, converge to the solution of a nonlinear initial value problem. But now, with linear differential equations, the two-point boundary value problem is tractable. If the known final coordinates and the final time are fixed, the principle of superposition may be applied to equations (3.2.13) to satisfy the boundary conditions represented by equations (3.2.2) and (3.2.3). For convenience rewrite equation (3.2.13) for the general (n + 1) iteration as

$$\dot{\underline{y}} = A\underline{y} + B \tag{3.2.14}$$

where

$$A = \left[\frac{\partial \underline{F}}{\partial \underline{y}}\right]_n \tag{3.2.15}$$

$$B = \underline{F}(\underline{y}^n) - \left[\frac{\partial \underline{F}}{\partial \underline{y}}\right]_n \underline{y}^n \tag{3.2.16}$$

At this point it is convenient to consider that the elements of $\underline{y}$ (and, of course, $\underline{F}$) have been ordered so that the p variables having known initial conditions are the first elements of $\underline{y}$ and the (2n - p) states having unknown initial conditions are the last elements of $\underline{y}$. This is in no way necessary. It is simply notationally and computationally expedient.

The homogeneous part of equation (3.2.14) is

$$\dot{\underline{y}} = A\underline{y} \tag{3.2.17}$$

Consider integrating this equation forward (2n - p) times with the initial vectors chosen as follows: For each integration, each element of the initial vector is zero except for one. For the first integration, element p + 1 is chosen to be 1. For the n^{th} integration, the (p + n)-th element is chosen to be 1. The reason for using these particular initial vectors to generate independent solutions to equation (3.2.17) will become apparent later.

Let H be a matrix of resulting homogeneous solutions whose general element a_{ki} is $y_i(t)$ generated with initial starting vector $\underline{y}_{o_k}$ defined above. In other words, the k-th column of H is the solution vector $\underline{y}(t)$ corresponding to initial vector $\underline{y}_{o_k}$. H has (2n - p) columns and 2n rows. Since equation (3.2.14) is linear, the principle superposition applies, and a general solution to (3.2.14) can be written as the sum of a particular solution to the non-homogeneous equation and (2n - p) linearly independent solutions of the homogeneous equation. In particular

$$\underline{y}(t) = H(t)\underline{c} + \underline{x}(t) \tag{3.2.18}$$

where $\underline{c}$ is a vector of arbitrary constants and where $\underline{x}(t)$ is any particular solution of (3.2.14), which satisfies initial constraints of equation (3.2.2). In particular let $\underline{x}(t) = \underline{y}^{n+1}(t)$, the most recent estimate for the optimal solution. At the final point this becomes

$$\underline{y}_f = H(t_f)\underline{c} + \underline{x}_f \tag{3.2.19}$$

Equation (3.2.19) represents 2n equations in the (2n - p) arbitrary c's. However, p of these equations are of no interest since they involve variables which are not constrained. Form the vectors $\underline{y}_f'$ and $\underline{x}_f'$ by deleting the p elements corresponding to states not involved in the final constraints. Form $H'(t_f)$ by deleting corresponding rows of H_f. H' is a (2n - p) by (2n - p) matrix. Then the pertinent equations may be written

$$\underline{y}_f' = H'(t_f)\underline{c} + \underline{x}_f' \tag{3.2.20}$$

The elements of $\underline{c}$ in equation (3.2.20) can now be selected to satisfy the required final boundary conditions.

$$\underline{c} = [H'(t_f)]^{-1} [\underline{y}'_f - \underline{x}'_f] \qquad (3.2.21)$$

The corresponding initial conditions are found by evaluating (3.2.18) at t_o and substituting $\underline{c}$ from equation (3.2.21). Note that the first p rows of $H(t_o)$ are all zero, while the last (2n - p) rows and columns form an identity matrix. Thus, the vector $\underline{c}$ does not affect known initial conditions and equation (3.2.18) becomes simply

$$y_{io} = c_i + x_{io} \qquad i = p+1, p+2, \ldots, 2n \qquad (3.2.22)$$

where the c_i's are given by equation (3.2.21). The advantage of choosing the initial vectors as described following equation (3.2.17) is now apparent. The y_{io} values are referred to as the updated initial values, and the c_i values are referred to as the updates.

If equation (3.2.14) is integrated forward, with the updated initial values from equation (3.2.22), the desired final endpoints will be approached. Under appropriate circumstances, the resulting endpoints will be closer in a Euclidian sense to the desired final endpoint than that of the old solution.

A complete iteration in solving the boundary value problem has now been made, and the entire process can be repeated again as follows:

(1) For a given set of initial values, repeatedly evaluate new values for y^{n+1} using equation (3.2.13) until the solution of the linear differential equation (3.2.14) is a close approximation to the solution of the nonlinear differential equation (3.2.1).

(2) Form the H matrix by integrating equation (3.2.17) with the appropriate initial vectors. Use the resulting final values to update the initial unknown values using equations (3.2.21) and (3.2.22).

(3) Repeat until the boundary values and the solution are within specified tolerances.

Note that step (1) may require 5 to 10 integration iterations. Experience indicates that it is unnecessary to obtain an accurate approximation to the nonlinear differential equations before updating the initial values. In practice, step (2) is performed after every integration of equations (3.2.14) and (3.2.17), except when two consecutive particular solutions differ considerably. Consequently, the convergence on the boundary conditions is simultaneous with the convergence on the solution.

3.2.3 Variable Final Point Boundary. The method for variable endpoints is essentially the same as that for fixed endpoints. Only a modification in obtaining the initial value updates $\underline{c}$ need be made.

Two methods have been suggested for handling such problems with an unspecified final time. Long (1965) suggests a change in independent variable, $t = as$, where s is the new independent variable with a range of $0 \leq s \leq 1$. The variable a is appended to the system as a pseudo-state variable whose derivative with respect to s is zero. The method works well. However, a relatively complex term corresponding to the new state variable, a, must be added to each differential equation. A second method has been developed by Lewallen (1967) and is the method followed below.

If the final time is free, there must be $2n + 1$ boundary conditions to completely specify the solution. If, as before, there are p initial coordinate constraints, there must be $(2n + 1 - p)$ constraints involving the final point. Let these be represented by

$$\psi_i(\underline{y}_f') = \psi_{if}^* \qquad i = 1,2,\ldots,2n+1-p \qquad (3.2.23)$$

or

$$\underline{\psi}(\underline{y}_f') = \underline{\psi}_f^*$$

where $\underline{\psi}_f^*$ represents a constant vector of desired values for the final constraints. If the terminal constraints (3.2.23) are not satisfied, the difference between the desired value $\underline{\psi}_f^*$ and the actual value $\underline{\psi}(\underline{x}_f')$ is called the terminal constraint dissatisfactions $\Delta\underline{\psi}$:

$$\Delta\underline{\psi} = \underline{\psi}_f^* - \underline{\psi}(\underline{y}_f') \qquad (3.2.24)$$

The terminal constraint dissatisfactions can be linearly approximated by summing

(1) the change in $\underline{\psi}$ due to small change in its arguments $\Delta\underline{y}_f'$, assuming the final time fixed and

(2) the change in $\underline{\psi}$ due to a change in the final time Δt_f assuming the final state fixed.

Symbolically this is

$$\Delta\underline{\psi} = \left[\frac{\partial\underline{\psi}}{\partial\underline{y}_f'}\right]\Delta\underline{y}_f' + \left.\frac{d\underline{\psi}}{dt}\right|_f \Delta t_f \qquad (3.2.25)$$

An equation has already been obtained from which $\Delta\underline{y}_f'$ can be expressed. Rewriting equation (3.2.20) gives

$$\Delta\underline{y}_f' = \underline{y}_f' - \underline{x}_f' = H'(t_f)\underline{c} \qquad (3.2.26)$$

Substituting (3.2.26) into (3.2.25) yields

$$\Delta\underline{\psi} = \left[\frac{\partial\underline{\psi}}{\partial\underline{y}_f'}\right] H'(t_f)\underline{c} + \left.\frac{d\underline{\psi}}{dt}\right|_f \Delta t_f \qquad (3.2.27)$$

The values for $\left.\frac{d\underline{\psi}}{dt}\right|_f$ can be computed directly from the values of $\left.\frac{dy'}{dt}\right|_f$ using the differential equality

$$\left.\frac{d\underline{\psi}}{dt}\right|_f = \left[\frac{\partial\underline{\psi}}{\partial\underline{y}_f'}\right]\left.\frac{dy'}{dt}\right|_f \qquad (3.2.28)$$

The values for the vector $\left.\frac{d\underline{y}'}{dt}\right|_f$ are available as a result of the last step taken in the numerical integration of the particular solution. Substituting equation (3.2.28) into equation (3.2.27) and simplifying yields

$$\Delta\underline{\psi} = \frac{\partial\underline{\psi}}{\partial\underline{y}_f'}\left[H'(t_f)\underline{c} + \left.\frac{d\underline{y}'}{dt}\right|_f \Delta t_f \right] \tag{3.2.29}$$

In practice, the column vector $\left.\frac{d\underline{y}'}{dt}\right|_f$ is adjoined to the matrix $H'(t_f)$ and equation (3.2.29) is written

$$\Delta\underline{\psi} = \left[\frac{\partial\underline{\psi}}{\partial\underline{y}_f'}\right]\left[H'(t_f) \;\vdots\; \frac{d\underline{y}'}{dt}\right]\underline{c}' \tag{3.2.30}$$

where

$$\underline{c}' = \begin{vmatrix} c_1 \\ c_2 \\ \vdots \\ c_{q-1} \\ \Delta t_f \end{vmatrix}$$

The updates for the initial values, $\underline{c}$ and Δt_f, can be solved for using equation (3.2.30) by a simple matrix multiplication and matrix inversion.

3.3 An Example of the Modified Newton-Raphson Method

As an example of theory developed in the previous section, consider the solution of the two-point boundary value problem arising from the optimal orbital mechanics problem posed in Chapter 5. To apply the modified

Newton-Raphson method, the set of differential equations to be integrated must first be linearized. In terms of the theoretical development, the elements of matrix A and vector B, defined by equations (3.2.15) and (3.2.16), must be determined. For illustration, consider the $\dot{u}$ equation, equation (5.1.50), which is repeated below for convenience.

$$\dot{u} = \frac{F}{M} \frac{\lambda_u}{\sqrt{\left(\frac{\lambda_g}{u}\right)^2 + \lambda_u^2}} - \frac{1}{R^2} \sin g = f_1 \tag{3.3.1}$$

The elements of the vector $\underline{y}$ for this problem are $[u, g, R, \lambda_u, \lambda_g, \lambda_r]^T$. According to equation (3.2.15) the partial derivatives of f_1 with respect to the elements of $\underline{y}$ form the first row of the A matrix. These derivatives are listed below.

$$A_{11} = \frac{\partial f_1}{\partial y_1} = \frac{\partial f_1}{\partial u} = \frac{F}{M} \frac{\lambda_u \lambda_g^2}{u^3} D^{-3/2} \tag{3.3.2}$$

$$A_{12} = \frac{\partial f_1}{\partial y_2} = \frac{\partial f_1}{\partial g} = -\frac{1}{R^2} \cos g \tag{3.3.3}$$

$$A_{13} = \frac{\partial f_1}{\partial y_3} = \frac{\partial f_1}{\partial R} = \frac{2}{R^3} \sin g \tag{3.3.4}$$

$$A_{14} = \frac{\partial f_1}{\partial y_4} = \frac{\partial f_1}{\partial \lambda_u} = \frac{F}{M} \frac{\lambda_g^2}{u^2} D^{-3/2} \tag{3.3.5}$$

B vector are exhibited in FORTRAN in subroutine DIFFL of Appendix B.

Next, the boundary conditions must be linearized. To do this the elements of the matrix $\left[\frac{\partial \underline{\psi}}{\partial \underline{y}_f'}\right]$ must be determined. For the optimal orbital mechanics problem, $\underline{y}_f' = [u_f, g_f, R_f, \lambda_{uf}, \lambda_{gf}, \lambda_{rf}]^T$, since all of the elements of $\underline{y}_f$ are involved in the constraints. Following the notation used in Appendix B, this matrix is called HPAR. The constraint equation (5.1.29) is repeated below as an example.

$$\psi_1 = \frac{u_f^{\,2}}{2} - \frac{1}{R_f} - C_E \qquad (3.3.11)$$

The elements of the first row of HPAR are given below.

$$\mathrm{HPAR}_{11} = \frac{\partial \psi_1}{\partial y_{1f}'} = \frac{\partial \psi_1}{\partial u_f} = u_f \qquad (3.3.12)$$

$$\mathrm{HPAR}_{12} = \frac{\partial \psi_1}{\partial y_{2f}'} = \frac{\partial \psi_1}{\partial g_f} = 0 \qquad (3.3.13)$$

$$\mathrm{HPAR}_{13} = \frac{\partial \psi_1}{\partial y_{3f}'} = \frac{\partial \psi_1}{\partial R_f} = \frac{1}{R_f^{\,2}} \qquad (3.3.14)$$

$$\mathrm{HPAR}_{14} = \mathrm{HPAR}_{15} = \mathrm{HPAR}_{16} = 0 \qquad (3.3.15)$$

All the elements of the HPAR matrix are exhibited in FORTRAN in subroutine BOUND of Appendix B. Once the preliminary work of linearization has been performed, the generalized subroutines exhibited in Appendix B can be used directly.

Subroutine QUASI controls the generation of a solution to a two-point boundary value problem. Although the listing of this subroutine in Appendix B is intended to be self-explanatory, a general description of the computational procedure is in order. Each generalized Newton-Raphson iteration requires the generation of a particular solution and $(2n - p)$ homogeneous solutions to equation (3.2.14). Since each solution has $2n$ independent variables, one iteration requires the integration of $2n(2n - p + 1)$ first order differential equations. Each of these solutions could be generated separately. However, if rapid access memory is limited, this method is computationally inefficient since matrix $A(t)$ must be evaluated $(2n - p + 1)$ times. Instead, all $(2n - p + 1)$ solutions are generated simultaneously. At any given time in the integration, matrix A need only be evaluated once in order to integrate all solutions forward one step. This method also saves considerable time in computational overhead.

Before solving the two-point boundary value problem, an estimate of the initial solution is stored in matrix XOLD at discrete times differing by a constant increment Δ. XOLD is defined so that $XOLD(i,1) = y_i(t_o)$ and $XOLD(i,j) = y_i(t_o + (j-1)\Delta)$.

A summary of the operation of subroutine QUASI is given below.

(1) Set the elements of matrix A and matrix B which are not functions of time to their initial values (call DIFFI).

(2) Set up initial vector X of length 2n(2n - p + 1) to be integrated. The elements of X correspond to the initial vectors for the particular solution and (2n - p) homogeneous solutions needed.

(3) Integrate all equations to the final time estimate. After each integration step, store the new particular solution in XOLD. The old particular solution, which has just been used to evaluate A and B, is of no further use and is destroyed.

(4) Evaluate the updates to the unknown initial values and to the final time estimate (call BOUND).

(5) Change the unknown initial conditions and final time by a fraction of the nominal updates. The fraction used depends upon certain convergence and stability criterion.

(6) Determine if the solution has converged to within desired tolerances. If not, return to step 3.

The actual integration in step 3 is performed by subroutine RUNKUT which calls subroutine DLSUB to evaluate the right hand side of the first order differential equations being integrated. DLSUB in turn calls subroutine DIFFL, which evaluates matrices A and B using the old particular solution stored in XOLD. Using A and B, subroutine DLSUB then evaluates the right hand sides of the particular and all homogeneous differential equations and returns these results to RUNKUT. Subroutine RUNKUT integrates forward one step and returns control subroutine QUASI, which stores the new solution in XOLD and determines if the final time has been encountered. This procedure is repeated until the final time is encountered. The details of these programs are described in the FORTRAN listings of Appendix B.

CHAPTER 4

THRUSTING HARMONIC OSCILLATOR

This chapter deals with the occurrence of multiple stationary solutions for problems having bounded control and periodic solutions. Such multiple solutions have been observed in a nonlinear orbital mechanics problem discussed in Chapter 5. In order to gain some insight into the latter problem, it is desirable to investigate first a simple linear problem which exhibits similar multiple solutions.

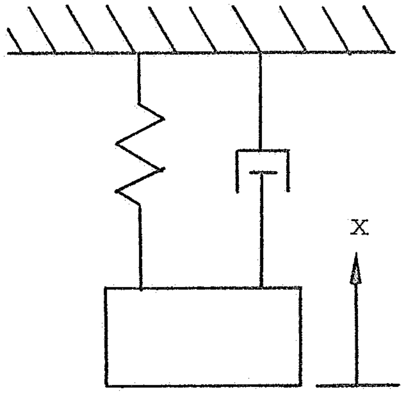

Fig. 4.1 Thrusting Harmonic Oscillator

Consider a mass m which is connected vertically to an inertial reference by a spring and dash pot as shown in Figure 4.1. The spring constant per unit mass is represented by k, and the coefficient of damping per unit mass is represented by c. The mass is capable of producing a bounded thrust per unit mass of T in the upward direction only. The mass is assumed constant. The equations of motion for this thrusting harmonic oscillator are

$$\dot{v} = T - cv - kx \qquad k > 0,\ c > 0 \tag{4.1.1}$$

$$\dot{x} = v \tag{4.1.2}$$

$$0 \leq T \leq T_{max} \tag{4.1.3}$$

where

x is the position of the mass measured in an upward direction from its equilibrium position

v is the mass velocity.

The problem is to transfer the mass from rest to a given final height h while minimizing the control effort,

$$J = \int_{t_o}^{t_f} T dt \tag{4.1.4}$$

The boundary conditions (algebraic equations of constraint) for this problem are

$$x_o = 0 \qquad v_o = 0 \qquad t_o = 0 \tag{4.1.5}$$

$$x_f = h \qquad v_f \text{ is free} \qquad t_f \text{ is free} \tag{4.1.6}$$

The Hamiltonian and G functions may be written as

$$H = \lambda_v(F - cv - kx) + \lambda_x v - F$$

$$= (\lambda_v - 1)F - \lambda_v(cv + kx) + \lambda_x v \tag{4.1.7}$$

$$G = \mu_1(x_f - h) \tag{4.1.8}$$

The initial point constraints have not been included in the sum G above, since doing so provides no useful information. Since H is not an explicit function of time, the first integral implies that H is equal to a constant. The adjoint variable Euler-Lagrange equations are

$$\dot{\lambda}_v = \lambda_v c - \lambda_x \tag{4.1.9}$$

and

$$\dot{\lambda}_x = \lambda_v k \tag{4.1.10}$$

The control switching function is seen from equation (4.1.7) to be

$$S = \lambda_v - 1$$

Therefore, the Maximum Principle (see section 1.3.5) yields the following control algorithm:

$$\begin{array}{ll} T = 0 & \lambda_v - 1 < 0 \\ 0 < T < T_{max} & \lambda_v - 1 \equiv 0 \\ T = T_{max} & \lambda_v - 1 > 0 \end{array} \qquad (4.1.11)$$

For singular control, that is, intermediate thrust, λ_v must equal 1, and all the time derivatives of λ_v must be 0. If this is the case, equation (4.1.9) requires $\lambda_x = c$; therefore, $\dot{\lambda}_x$ is zero and equation (4.1.10) requires $k = 0$. This contradicts the assumptions of the problem and singular control is ruled out. The control must be bang-bang.

Solving equations (4.1.9) and (4.1.10) for λ_v gives

$$\lambda_v = e^{\frac{c}{2}t}(A \sin bt + B \cos bt), \qquad (4.1.12)$$

if

$$c^2 - 4k < 0$$

where

$$b = \frac{1}{2}(4k-c^2)^{1/2} \qquad (4.1.13)$$

The case of $c^2 - 4k \geq 0$ is not of interest since the physical system does not exhibit periodic oscillations under these circumstances.

The possibility of null thrust

$$T(0^+) = 0 \qquad (4.1.14)$$

for the initial control will not be considered since the state would remain unchanged. Starting with maximum thrust it is seen from equation (4.1.7) with H(0) = 0 (to be shown) that

$$\lambda_v(0) = \lambda_{vo} = 1 \qquad (4.1.15)$$

In order to maintain $T = T_{max}$ at the next instant of time,

$$\dot{\lambda}_v(0) = \dot{\lambda}_{vo}, \text{ a positive constant} \qquad (4.1.16)$$

The function λ_v is shown in Figure 4.2 for a typical set of parameters. Note that the envelope of maximum points is asymptotically increasing. The corresponding control is shown in Figure 4.2b. Notice that $T = T_{max}$ for $\lambda_v > 1$, and $T = 0$ for $\lambda_v < 1$. The control is seen to be a series of thrusting intervals separated by coasting or null thrust intervals. Each thrusting interval is longer than the previous one. However, since the period P of λ_v is fixed, no thrusting interval can be longer than the P/2. The transversality conditions require that

$$\lambda_{vf} = 0 \qquad (4.1.17)$$

$$H_f = 0. \qquad (4.1.18)$$

The former condition is the cutoff condition and the latter condition requires that H = 0 throughout. If these conditions are satisfied, equation (4.1.7) gives

$$\lambda_{xf} v_f - F_f = 0 \qquad (4.1.19)$$

Since $\lambda_{vf} = 0$ by equation (4.1.17), it can be seen from equation (4.1.11) that

$$F_f = 0 \qquad (4.1.20)$$

Since $\dot{\lambda}_{vf}$ is always negative (see Figure 4.2), equations (4.1.17) and (4.1.9) determine that $\lambda_{xf} \neq 0$. Thus, equations (4.1.19) and (4.1.20) require that

$$v_f = 0 \qquad (4.1.21)$$

Although analytic solutions can be obtained, the evaluation of the boundary conditions becomes tedious for solutions with multiple thrusting intervals. The nature of the solutions is easily observed with the aid of numerical integration. A parametric flooding technique (see section 3.1.1) is practical, since solution curves are members of a one parameter family of curves with parameter λ_{xo}, or equivalently, $\dot{\lambda}_{vo}$. That is, all possible solutions can be generated by continuously varying $\dot{\lambda}_{vo}$ from 0 to $+\infty$.

With a value chosen for $\dot{\lambda}_{vo}$, differential equations (4.1.1), (4.1.2), (4.1.9) and (4.1.10) are

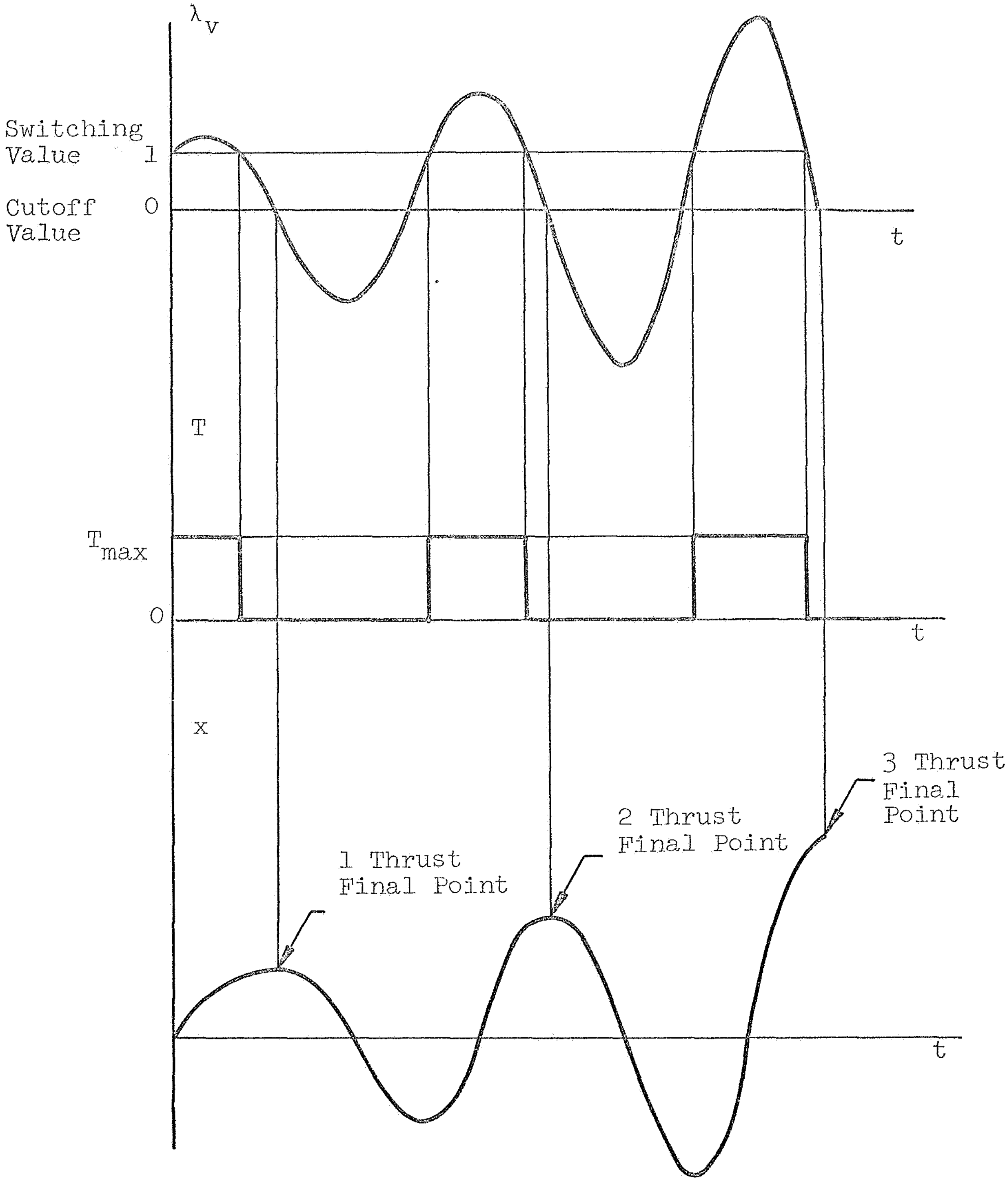

Fig. 4.2 The Cutoff Function, Thrust Magnitude and Height

integrated until one of the zeros of the cutoff function, equation (4.1.17), is encountered. As described in section 1.3.4, the zeros of the cutoff function represent points satisfying the transversality conditions.

Since v is obviously a periodic function for $c^2 - 4k < 0$, equation (4.1.21) implies that the cutoff function is satisfied during each period of oscillation. Since a thrusting interval occurs during each period, solutions exist with any number of thrusting periods.

The existence of multiple stationary solutions is easy to observe. Limits exist on the range of positive heights that may be reached by a solution with any specified number of thrusting intervals. For a solution with two thrusting intervals, the minimum specified height which can be optimally reached may be found by setting $\dot{\lambda}_{vo} = \epsilon$ and numerically integrating until the third zero of the cutoff function λ_v occurs. It can be seen from Figure 4.2 that as ϵ approaches 0, the length of the first thrusting interval approaches 0. On the other hand, as $\dot{\lambda}_{vo}$ approaches $+\infty$, the length of the initial thrusting period approaches P/2, the maximum allowable. In this way the range of achievable positive heights can be computed.

Figure 4.3 shows the range of heights which can be reached for solutions with up to seven thrusting

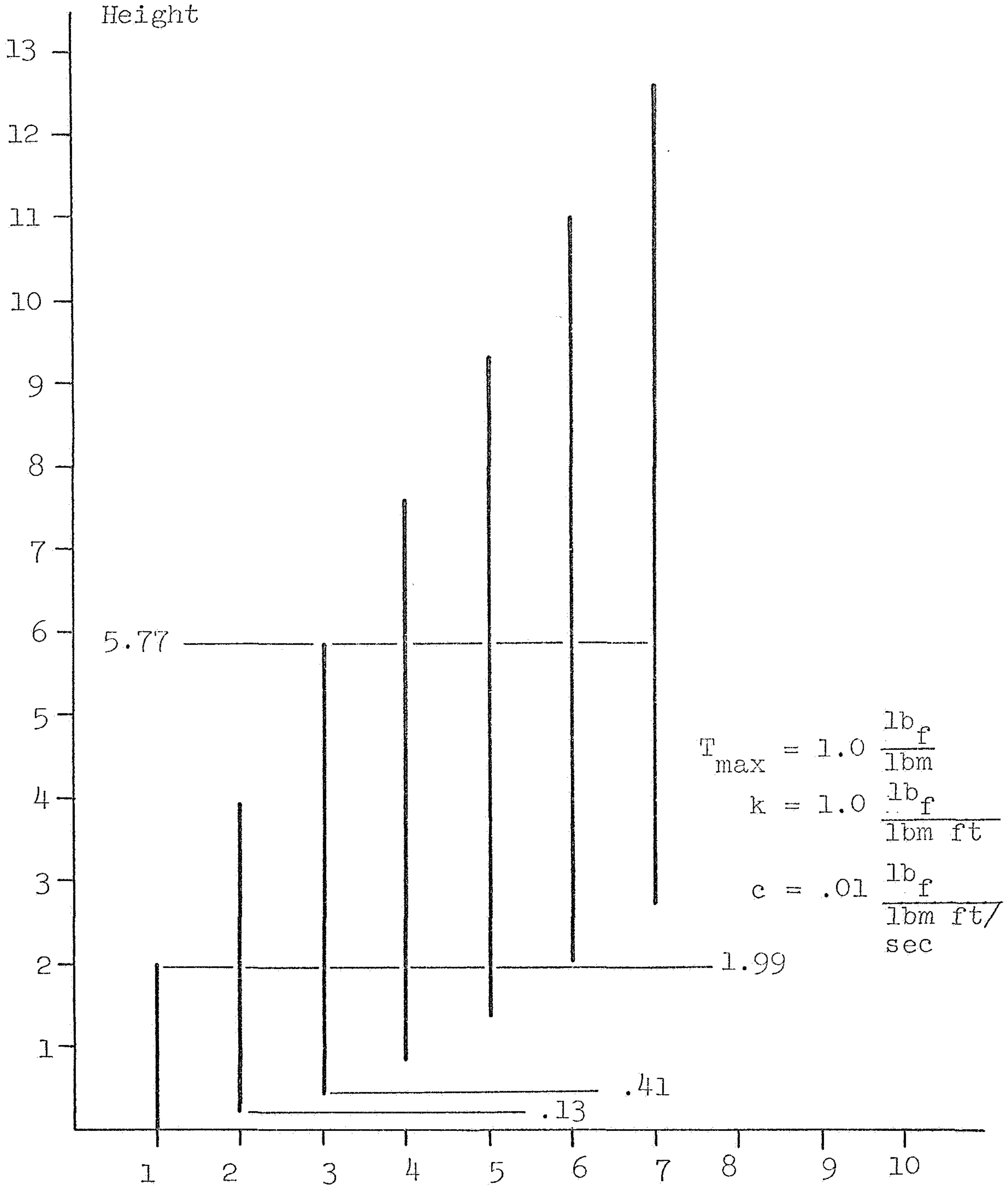

Fig. 4.3 Attainable Heights

intervals. Note that all heights between .13 ft. and some maximum attainable height can be reached in more than one way. Heights between .41 and 1.99 ft. can be reached by solutions with one, two, or three thrusting periods. Each of these solutions is a stationary solution satisfying the first three necessary conditions of section 1.2. Here is a clear case of multiple stationary solutions. The number of stationary solutions to a given height increases as the given height increases. From Figure 4.3 it can be seen that a height of 5.77 ft. can be reached with from three to seven thrusting intervals.

Although each of these solutions satisfies necessary conditions for optimality, Figure 4.4 shows that the effort required to achieve a given height is different for each mode of transfer. Only one global optimum exists in each case. To reach a height of 5.77 ft. requires an effort of 9.42 lb_f-sec. using three thrusting intervals. By using the optimal four thrusting intervals, the effort can be reduced to 8.36 lb_f-sec. Five thrusting intervals requires an effort of 8.80 lb_f-sec.

Although not verified analytically, it is evident that each solution to a given height satisfies the endpoint sufficiency condition developed in Chapter 2. That is, each solution represents a local minimum of effort with respect to small variations in the final

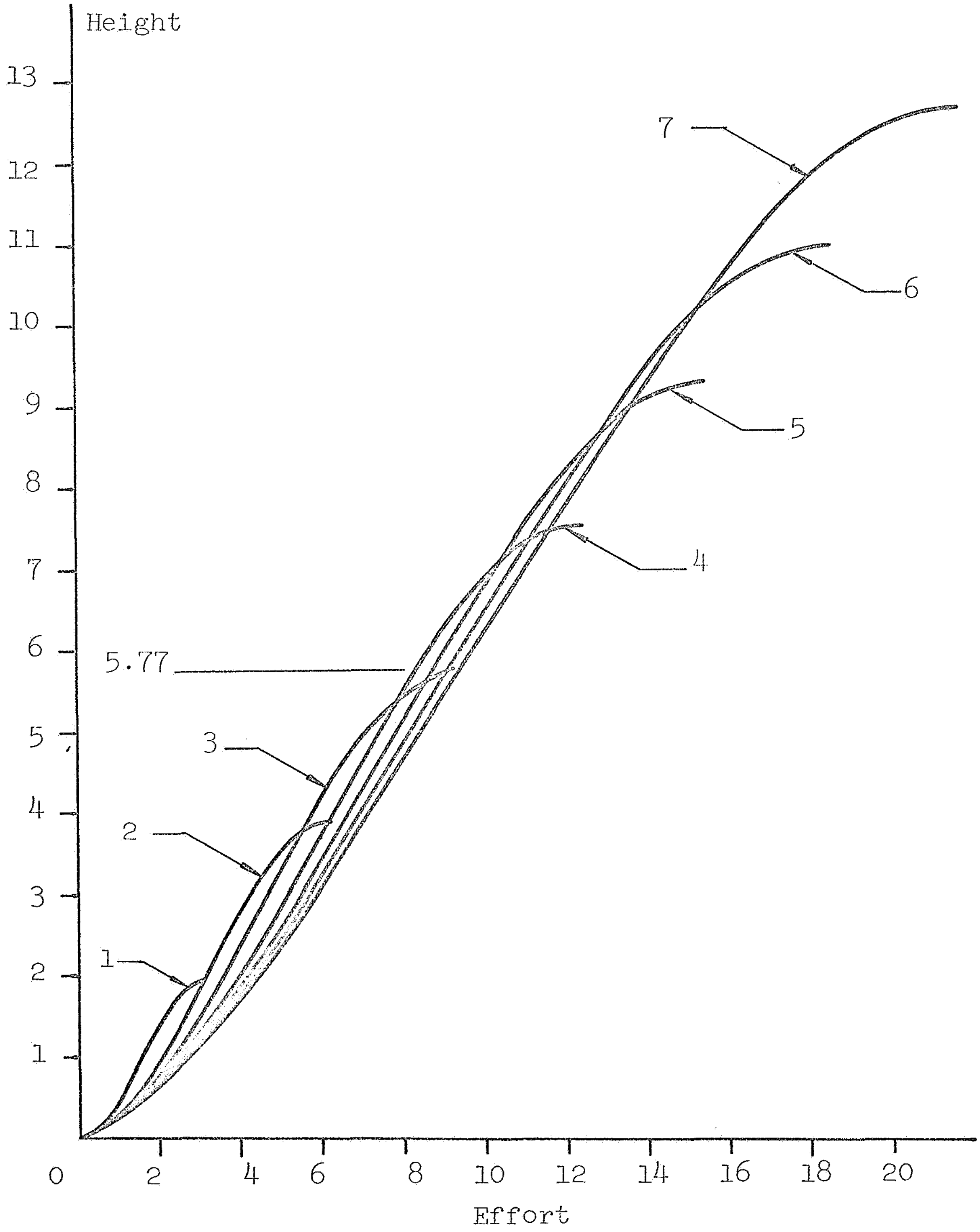

Fig. 4.4 Effort Required

velocity. This conclusion may be drawn from energy considerations. From an energy standpoint, the oscillator will gain the maximum amount of energy by centering each thrusting period about the maximum positive oscillator velocity. Therefore, if the length of each thrusting period is fixed at the optimum value, but the time of thrust initiation is changed slightly, the oscillator will gain less than optimum energy from each thrust and consequently attain a less than optimum height.

Without further sufficiency conditions, no criteria exist for selecting the global minimum for this problem. There seems to be little hope for the development of global sufficiency conditions from the calculus of variations in its current formulation. The reason for this is apparent. The necessary and sufficient conditions that have been developed to date are effective only in determining trajectories which are local optimums. That is, only small variations in the state variable trajectory are considered. It is clear that the state trajectory for a two thrust control program is not a small variation from the state trajectory for a single thrust program.

Referring to similar multiple thrusting impulses involved in optimal orbital transfers, D. F. Lawden (1963, pp. 112-114) concludes the following:

> . . . the conditions we have found for an optimal trajectory serve only to identify maneuvers which are optimal relative to small variations of the thrust program. This implies that a number of such optimal trajectories may be available in any particular case and we have no criterion, other than direct comparison of the characteristic velocities [performance index] of each, for deciding which represents the absolute optimal mode of transfer . . . .
>
> . . . modes employing two or more impulses and satisfying all of the [necessary] conditions may also be found and these, also, will represent relative optima. It is clear that the theory, in its present state, requires that we should first choose the order in which the various thrust phases shall succeed one another, and our [necessary] conditions will then select the relative optima from the class of all programs for which the thrust sequence follows the chosen pattern. No criteria is known for which the optimal pattern can be ascertained beforehand.

CHAPTER 5

AN OPTIMAL ORBITAL MECHANICS PROBLEM

This chapter considers the problem of transferring a low thrust space vehicle from an initial circular orbit to any coplanar elliptic orbit of given energy and angular momentum. The problem is to determine the thrust program so as to minimize

$$J = \int_{t_o}^{t_f} T(t)\, dt$$

where T(t) is the magnitude of the thrust. A typical transfer is shown in Figure 5.1. The final argument of periapsis ω_f, the final true anomoly φ_f, the final range angle f_f, and the final time t_f are all unspecified and considered free. For such transfers from an initial circular orbit, the orientation of the final orbit is of no concern, since once ω_f is known, any desired orientation can be optimally achieved by simply coasting in the initial circular orbit for an appropriate period of time.

Many investigators have turned their attention to optimal orbital transfers in the past two decades. Bell (1968) gives an excellent survey of problems which have been investigated, including a bibliography with

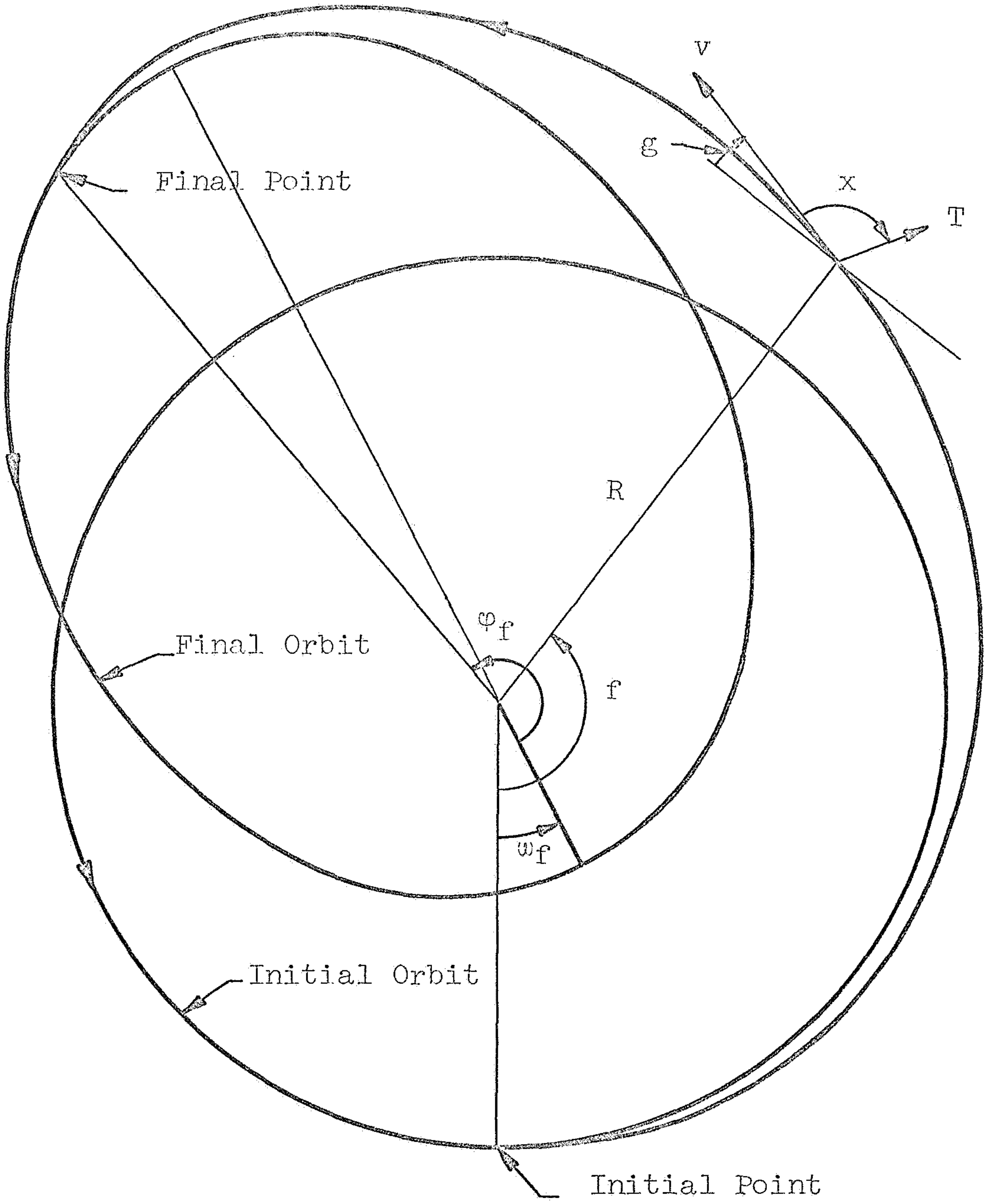

Fig. 5.1 Typical Orbital Transfer

160 entries. Recent investigations concerning the minimum fuel problem have followed two distinct approaches. In the first approach, simplifying assumptions are made which allow analytical treatment of some problems.

Lawden (1963), Robbins (1965), and others have analytically treated minimum fuel transfers using an impulsive approximation. McIntyre and Crocco (1966; 1967) have examined minimum fuel transfers between circular orbits under the condition that the separation of the orbits is small and have found analytic approximations exhibiting multiple thrusting periods.

Problems concerning minimum fuel orbital transfers with low thrust have usually been investigated using a computational approach. Numerical studies of this problem have been made by McCue (1967), Handelsman (1966), and McCue and Bender (1965). Both analytic and numeric investigations have contributed substantially to the understanding of the problem of minimum fuel orbital transfers.

In the first section of this chapter, this particular problem is set up and the optimization conditions derived. An instance of multiple stationary solutions not noted by previous investigators is described in section 5.2. In section 5.3 the endpoint sufficiency condition for this problem is described and the computational procedure for implementing it is discussed. In section 5.4, several cases of multiple solutions are illustrated with trajectories in semi-major-axis-eccentricity space. The final section describes the general nature of the low thrust orbital transfers which were investigated.

5.1 Analysis

Using a natural coordinate system, the differential equations governing the dynamics of a space vehicle shown in Figure 5.1 are

$$\dot{v} = \frac{T}{m} \cos x - \frac{GM^*}{r^2} \sin g \tag{5.1.1}$$

$$\dot{g} = \frac{T}{mv} \sin x - \frac{GM^*}{r^2 v} \cos g + \frac{v}{r} \cos g \tag{5.1.2}$$

$$\dot{r} = v \sin g \tag{5.1.3}$$

$$\dot{m} = - \frac{T}{v_e} \tag{5.1.4}$$

$$\dot{f} = \frac{v}{r} \cos g \tag{5.1.5}$$

where

v = velocity magnitude

g = flight path angle (the angle from the local horizontal to the velocity vector)

r = radial distance from the gravitational force center to the vehicle

m = vehicle mass

f = range angle $(f(t_o) = 0)$

x = thrust vector control angle (the angle from the velocity vector to the thrust vector)

T = thrust magnitude

G = universal gravitational constant

M^* = mass of the gravitational force center; constant

v_e = engine exhaust velocity; constant

In deriving the above equations, only planar motion about a homogeneous, spherically symmetric central body has been considered. The vehicle has been assumed to be a point mass with mass much less than that of the gravitational force center. In addition, it is assumed that the thrust magnitude is bounded by

$$0 < T < T_{max} \tag{5.1.6}$$

It is useful to nondimensionalize equations (5.1.1) - (5.1.6) in order to compare orbital transfers about one central body with those about another. The non-dimensionalization is based upon the energy of the initial orbit E_o and variables associated with a circular orbit having the same energy.

For this purpose define the following constants

$$R_c = \frac{GM^*m_o}{2E_o} \tag{5.1.7}$$

$$v_c = \sqrt{\frac{GM^*}{R_c}} \tag{5.1.8}$$

$$a_c = \frac{GM^*}{R_c^{\ 2}} \tag{5.1.9}$$

$$t_c = \frac{R_c}{v_c} \tag{5.1.10}$$

$$E_c = \frac{E_o}{m_o} = - \frac{GM^*}{2R_c} \tag{5.1.11}$$

$$h_c = R_c v_c m_o \tag{5.1.12}$$

where

m_o = initial vehicle mass

a_c = acceleration constant

h_c = angular momentum constant

With these constants, the following nondimensional variables are defined

$$R = \frac{r}{R_c} = \text{nondimensional radius} \tag{5.1.13}$$

$$u = \frac{v}{v_c} = \text{nondimensional velocity} \tag{5.1.14}$$

$$\tau = \frac{t}{t_c} = \text{nondimensional time} \tag{5.1.15}$$

$$M = \frac{m}{m_o} = \text{nondimensional mass} \tag{5.1.16}$$

$$F = \frac{T}{m_o a_c} = \text{nondimensional thrust} \tag{5.1.17}$$

$$E = \frac{E}{E_c} = \text{nondimensional energy} \qquad (5.1.18)$$

$$h = \frac{h}{h_c} = \text{nondimensional angular momentum} \qquad (5.1.19)$$

Substituting equations (5.1.13) - (5.1.19) into the system differential equations yields the following nondimensional system equations:

$$u' = \frac{F}{M} \cos x - \frac{1}{R^2} \sin g \qquad (5.1.20)$$

$$g' = \frac{F}{Mu} \sin x - \frac{1}{R^2 u} \cos g + \frac{u}{R} \cos g \qquad (5.1.21)$$

$$R' = u \sin g \qquad (5.1.22)$$

$$m' = - \frac{F}{u_e} \qquad (5.1.23)$$

$$f' = \frac{u}{R} \cos g \qquad (5.1.24)$$

where the prime indicates differentiation with respect to the nondimensional time τ.

The problem to be investigated then is that of minimizing

$$J = \int_{\tau_o}^{\tau_f} F \, d\tau \qquad (5.1.25)$$

subject to differential constraints (5.1.20) - (5.1.24) and endpoint constraints

$$u_o - 1 = 0 \tag{5.1.26}$$

$$g_o = 0 \tag{5.1.27}$$

$$R_o - 1 = 0 \tag{5.1.28}$$

$$\frac{1}{2}u_f^2 - \frac{1}{R_f} - C_E = 0 \tag{5.1.29}$$

$$R_f u_f \cos g_f - C_h = 0 \tag{5.1.30}$$

where C_E and C_h are constants.

Constraints (5.1.26) - (5.1.28) specify an initial circular orbit. Equation (5.1.29) specifies that the energy of the final orbit is to be C_E and equation (5.1.30) specifies that the angular momentum of the final orbit is to be C_h. The final energy and angular momentum specify the shape of final orbit, but not the orientation.

To determine the necessary conditions for a minimum fuel orbital transfer, the H and G functions, equations (1.2.7) and (1.2.8), are first formed:

$$\begin{aligned} H = \lambda_u &[\frac{F}{M} \cos x - \frac{1}{R^2} \sin g] \\ &+ \lambda_g [\frac{F}{Mu} \sin x - \frac{1}{R^2 u} \cos g + \frac{u}{R} \cos g] \quad (5.1.31) \\ &+ \lambda_r [u \sin g] - \lambda_m \frac{F}{u_e} - F \end{aligned}$$

Equation (5.1.24) need not be included in H, since f is an auxiliary variable not directly coupled to the other equations. The G function is

$$G = \mu_1 \left[\frac{1}{2}u_f^2 - \frac{1}{R_f} - C_E\right] + \mu_2 \left[R_f u_f \cos g_f - C_h\right] \qquad (5.1.32)$$

Constraints (5.1.26) - (5.1.28) have not been included in the G function, since doing so would yield no useful information. Forming the adjoint variable Euler-Lagrange differential equations, equations (1.2.10), yield

$$\lambda_u' = \lambda_g \left[\frac{F}{Mu^2} \sin x - \frac{1}{R^2u^2} \cos g - \frac{1}{R} \cos g\right] \qquad (5.1.33)$$

$$- \lambda_r \sin g$$

$$\lambda_g' = \lambda_u \frac{\cos g}{R^2} - \lambda_g \left[\frac{1}{R^2u} - \frac{u}{R}\right] \sin g \qquad (5.1.34)$$

$$- \lambda_r u \cos g$$

$$\lambda_r' = - \frac{2\lambda_u}{R^3} \sin g - \lambda_g \left[\frac{1}{R^3u} - \frac{u}{R^2}\right] \cos g \qquad (5.1.35)$$

$$\lambda_m' = \lambda_u \frac{F}{M^2} \cos x + \lambda_g \frac{F}{M^2u} \sin x \qquad (5.1.36)$$

The control variable Euler-Lagrange equation, equation (1.2.11) for control x is

$$\frac{F}{M} \left[- \lambda_u \sin x + \frac{\lambda_g}{u} \cos x\right] = 0 \qquad (5.1.37)$$

For $F \neq 0$ equation (5.1.37) implies

$$\sin x = \frac{\lambda_g/u}{\sqrt{\left(\frac{\lambda_g}{u}\right)^2 + \lambda_u{}^2}} \tag{5.1.38}$$

and

$$\cos x = \frac{\lambda_u}{\sqrt{\left(\frac{\lambda_g}{u}\right)^2 + \lambda_u^2}} \tag{5.1.39}$$

Since the second control variable F is bounded and appears linearly in H, "bang-bang" control may be expected. The possibility of singular control described by Lawden (1963, pp. 86-94) has been determined to be nonoptimal, in general, by Kelly, Kopp, and Moyer (1967, pp. 92-100). Therefore, the case of the switching function $S \equiv 0$ is not considered here. By substituting equations (5.1.38) and (5.1.39) into equation (5.1.31) and simplifying the term multiplying F, the thrust switching function may be written

$$S = \sqrt{\left(\frac{\lambda_g}{u}\right)^2 + \lambda_u^2} - M\left(1 + \frac{\lambda_m}{u_e}\right) \tag{5.1.40}$$

The thrust control may be written as

$$\begin{aligned} F &= F_{max} & S &> 0 \\ F &= 0 & S &< 0 \end{aligned} \tag{5.1.41}$$

The transversality equations, equations (1.2.12) - (1.2.15), yield

$$\mu_1 u_f + \mu_2 R_f \cos g_f + \lambda_{uf} = 0 \qquad (5.1.42)$$

$$- \mu_2 R_f u_f \sin g_f + \lambda_{gf} = 0 \qquad (5.1.43)$$

$$\frac{\mu_1}{{R_f}^2} + \mu_2 u_f \cos g_f + \lambda_{rf} = 0 \qquad (5.1.44)$$

$$\lambda_{mf} = 0 \qquad (5.1.45)$$

$$H_f = 0 \qquad (5.1.46)$$

Eliminating μ_1 and μ_2 from equations (5.1.42) - (5.1.44) yields

$$\lambda_{gf} \left(\frac{u_f}{R_f} - \frac{1}{u_f {R_f}^2} \right) \cos g_f$$

$$+ (\lambda_{rf} u_f - \frac{\lambda_{uf}}{{R_f}^2}) \sin g_f = 0 \qquad (5.1.47)$$

Using equations (5.1.47) and (5.1.31), it is observed that condition (5.1.46) becomes

$$\sqrt{\left(\frac{\lambda_{gf}}{u_f} \right)^2 + \lambda_{uf}^2} - M_f \; (1 + \frac{\lambda_{mf}}{u_e}) = 0 \qquad (5.1.48)$$

Equations (5.1.45), (5.1.47) and (5.1.48) constitute a set of necessary conditions which the endpoints must satisfy. Comparing equation (5.1.40) to equation (5.1.48), it is evident that points which qualify as endpoints also qualify as switching points for the thrust magnitude control.

Imposing condition (5.1.45) on equation (5.1.48) yields a cutoff function.

$$C = \sqrt{\left(\frac{\lambda_{gf}}{u_f}\right)^2 + \lambda_{uf}^2} - M_f \tag{5.1.49}$$

The zeros of this function determine points at which numerical integration may be terminated. Substituting the optimal thrust direction control from equations (5.1.38) and (5.1.39) into the nondimensionalized state variable differential equations, (5.1.20) - (5.1.23) and adjoint variable differential equations, (5.1.33) - (5.1.36) gives

$$u' = \frac{F\lambda_u}{MD^{1/2}} - \frac{1}{R^2} \sin g \tag{5.1.50}$$

$$g' = \frac{F\lambda_g}{Mu^2D^{1/2}} + \left[\frac{u}{R} - \frac{1}{R^2u}\right] \cos g \tag{5.1.51}$$

$$R' = u \sin g \tag{5.1.52}$$

$$M' = \frac{F}{u_e} \tag{5.1.53}$$

$$\lambda_u' = \lambda_g \left[\frac{F\lambda_g}{Mu^3 D^{1/2}} - \left[\frac{1}{R^2 u^2} + \frac{1}{R} \right] \cos g \right] - \lambda_r \sin g \qquad (5.1.54)$$

$$\lambda_g' = \frac{\lambda_u \cos g}{R^2} - \lambda_g \left[\frac{1}{R^2 u} - \frac{u}{R} \right] \sin g - \lambda_r u \cos g \qquad (5.1.55)$$

$$\lambda_r' = \frac{2\lambda_u}{R^3} \sin g - \lambda_g \left[\frac{2}{R^3 u} - \frac{u}{R^2} \right] \cos g \qquad (5.1.56)$$

$$\lambda_m' = \frac{F}{M^2} D^{1/2} \qquad (5.1.57)$$

where

$$D = \left(\frac{\lambda_g}{u} \right)^2 + \lambda_u^2$$

The integration of this set of equations subject to equations (5.1.26) - (5.1.30) and (5.1.42) - (5.1.46) yields the desired optimal trajectories.

5.2 Existence of Multiple Stationary Solutions

The problem posed in the previous section permits the existence of multiple thrust solutions. Mason (1967, pp. 49-70) has shown the existence of two solutions to a problem as posed in section 5.1: one solution is characterized by a single continuous thrust to the desired endpoint; a second solution, composed of two thrusting periods separated by a null-thrust coasting arc, reaches the same endpoint with a different performance index. Both solutions satisfy the necessary conditions of the calculus of variations. This type of multiple stationary solution is similar to the type of multiple stationary solutions exhibited by the thrusting harmonic oscillator in Chapter 4. Because of this similarity, it is probable that both the single thrust and the thrust-coast-thrust transfers satisfy the endpoint sufficiency condition described in Chapter 2. Verification of this hypothesis is left to future investigations. Several other investigators have also observed optimal orbital transfers with multiple thrusting periods (McCue, 1967; Handelsman, 1966; McIntyre and Crocco, 1967).

The investigation pursued in this chapter will instead be limited to orbital transfers with a single thrusting period. That is, thrust magnitude F is not considered to be a control, but will be of fixed magnitude for the entire duration of the transfer, i.e.,

$$F = F_{max} \qquad t_o < t < t_f \qquad (5.2.1)$$

If F is not a control, the M equation, equation (5.1.53) is no longer coupled to the system of differential equations represented by equations (5.1.50) - (5.1.57). Instead, M is a known function of time independent of u, g, R, and x.

$$M = \frac{F_{max}}{u_e} (t - t_o) \qquad (5.2.2)$$

Therefore the M equation need not be adjoined to the system through the Lagrange multiplier λ_m. All equations in the previous section are explicitly independent of time and will remain valid if λ_m is set to zero and M is evaluated using equation (5.2.2).

Optimal trajectories satisfying equation (5.2.1) are determined by the six differential equations (5.1.50) - (5.1.52), and (5.1.54) - (5.1.56) subject to the seven boundary conditions represented by equations (5.1.26) - (5.1.30), (5.1.47) and (5.1.48).

Throughout this chapter, orbits will be displayed as points in a plane having coordinates of nondimensional semimajor axis A and eccentricity e. Except for the sense of the angular momentum vector, there is a unique mapping between the variables E and h, whose final values have been specified and the variables A and e. The relationships are given below

$$A = - \frac{1}{2E} \qquad (5.2.6)$$

$$e^2 = 1 - 2Eh^2 \qquad (5.2.7)$$

Here A is the nondimensional semimajor axis. The nondimensional variables E and h are defined by equations (5.1.18) and (5.1.19).

Before a solution to the boundary value problem was attempted, the method of flooding described in section 3.1 was used to generate a locus of optimal endpoints in A-e space. To give physical significance to the required initial guesses, the following equations were derived for a fixed mass vehicle which relate the initial unknown Lagrange multipliers to the initial value of the control angle x_o and its derivative:

$$\lambda_{uo} = \cos x_o \tag{5.2.3}$$

$$\lambda_{go} = u_o \sin x_o \tag{5.2.4}$$

$$\lambda_{ro} = \frac{1}{u_o \cos (g_o + x_o)} \left[- F_{max} \sin x_o - u_o x_o' + \frac{u_o^2}{R_o} \sin x_o \sin (g_o + x_o) + \frac{\cos g_o}{R_o^{\ 2}} \right] \tag{5.2.5}$$

The first two of these relations result from the control Euler-Lagrange equation (5.1.37), the initial conditions and the fact that $H_o = 0$. The last equation results from the implicit differentiation of the control variable Euler-Lagrange equation and subsequent elimination of the resulting time derivatives using the state and adjoint differential equations.

Figure 5.2 displays the locus of optimal endpoints reached by initiating integration with $x_o = 4^o$

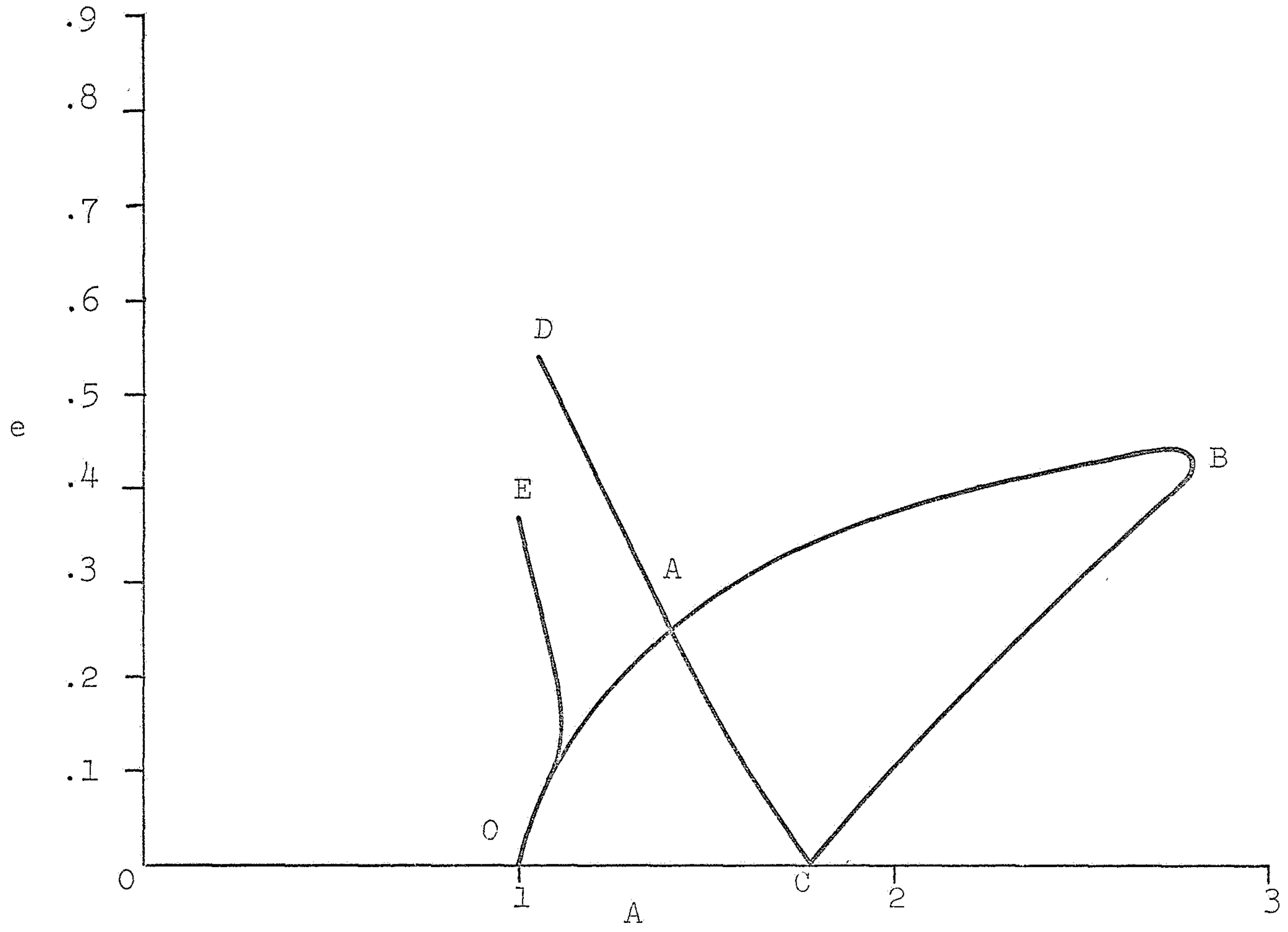

Fig. 5.2 Locus of Optimal Endpoints: $x_o = 4^o$

and varying values of x_o'. The coordinates used in the figure are semimajor axis and eccentricity. Each point on the locus was determined by terminating integration when the first zero of the cutoff function (5.1.49) was encountered. As x_o' was increased from a given initial value, the final point of the trajectory moved continuously forming the locus curve OABC. With a further increase in x_o' the locus was generated from C to D passing a second time through point A. A still further increase x_o' resulted in a jump from D to E and then a continuous return to O along curve EO. The fact that the locus of optimal endpoints intersects itself at A indicates that two solutions exist to point A.

The multiplicity of solutions is compounded if a second locus of optimal endpoint for $x_o = 16^o$ is superimposed on the one for $x_o = 4^o$ as shown in Figure 5.3. The movement of the optimal final point along the $x_o = 16^o$ locus for increasing x_o' is similar to that described for the $x_o = 4^o$ case. The 16^o locus not only intersects itself, as did the 4^o locus, but also intersects the 4^o locus in three places. Since similar loci exist for all intermediate angles, $4^o \leq x_o \leq 16^o$, it is evident that a complete region of endpoint space exists which can be reached by two distinct "optimal" trajectories.

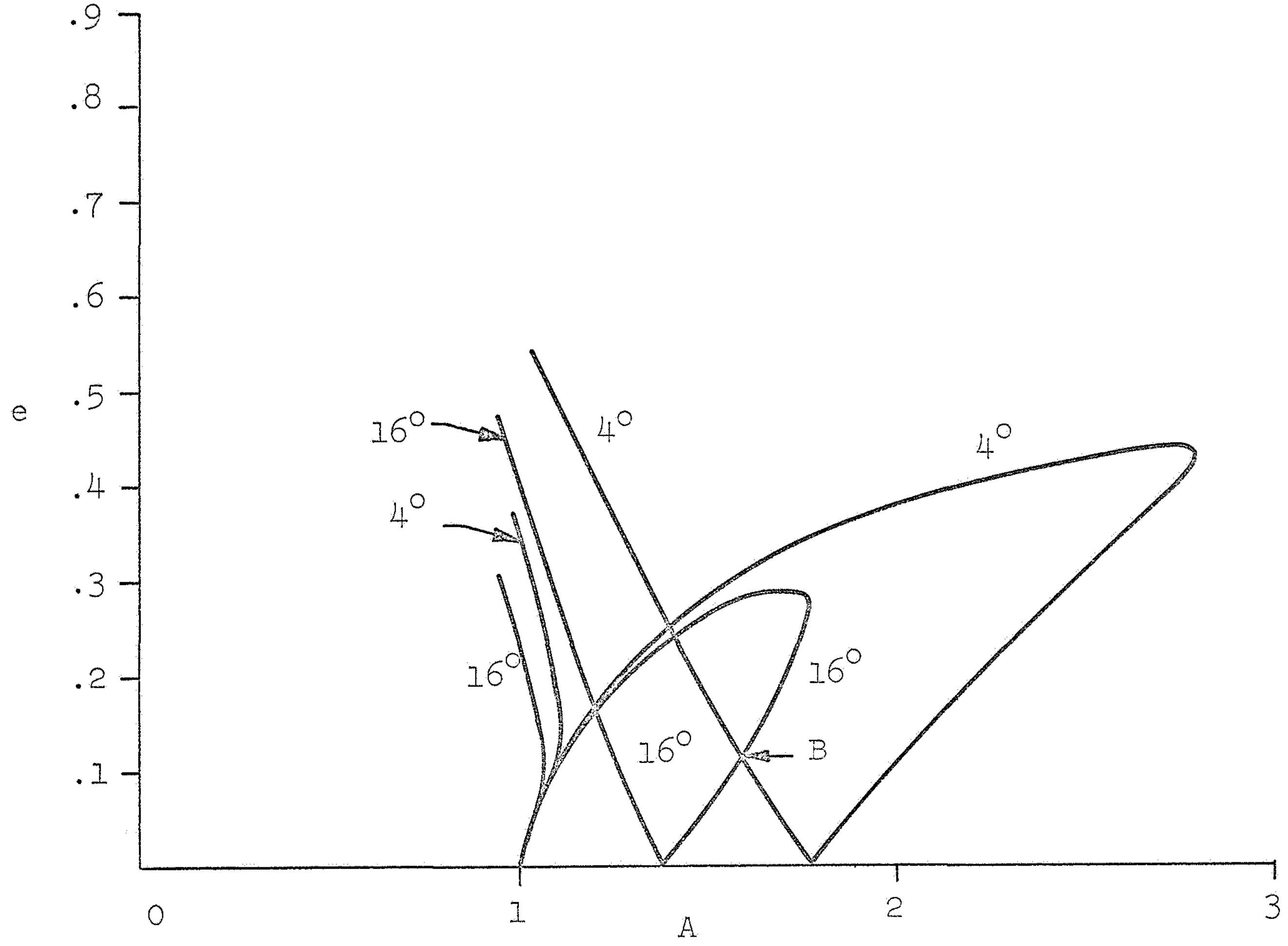

Fig. 5.3 Optimal Endpoints: $x_o = 4^\circ$ and 16°

5.3 Computational Procedures

The endpoint sufficiency test developed in Chapter 2 was applied to the case of multiple stationary solutions described in the previous section. Since the nonlinear differential equations describing the orbital transfer problem can not be integrated analytically, resort must be made to the numerical algorithm prescribed for such cases by section 2.6. This numerical algorithm assumes the user has the capability of solving two-point boundary value problems. Chapter 3 has set forth a method for solving such problems, complete with detailed references to allied computer programs listed in Appendix B. It remains for this section to describe in detail programs designed to implement the numerical algorithm proposed in section 2.6. In addition, the overall computational scheme for determining minimum fuel orbital transfers will be outlined, thus completing the description of programs contained in Appendix B.

5.3.1 The Endpoint Sufficiency Condition.

The endpoint sufficiency condition is given by equation (2.3.5). In this case the general endpoint is represented by the eight variables $(u_o, g_o, R_o, \tau_o, u_f, g_f, R_f, \tau_f)$. The endpoint constraints are represented by equations (5.1.26) - (5.1.30), and the following constraint on the initial time:

$$\tau_o = 0 \qquad (5.3.1)$$

Since there are six equations involving the eight endpoint variables, there are two degrees of freedom. That is, two endpoint variables involved in constraints which relate two or more endpoint variables (nontrivial constraints) may be independently specified, and the other six variables will then be given by the equations of constraint. Constraints which fix an endpoint coordinate simply result in a row of zeros in the Ω matrix of equation (2.3.5). Therefore, only the two nontrivial constraints, equations (5.1.29) and (5.1.30), and the nonfixed coordinates u_f, g_f, r_f and τ_f will be considered in the following analysis.

Any two of these variables can be selected as the independent variables. For convenience, r_f and τ_f are chosen as the independent variables.

In terms of the notation of section 2.3, the pertinent relations are

$$\left.\begin{aligned} &\underline{v} = \begin{vmatrix} R_f \\ \tau_f \end{vmatrix} \qquad \underline{w} = \begin{vmatrix} u_f \\ g_f \end{vmatrix} \qquad \underline{r} = \begin{vmatrix} u_f \\ g_f \\ R_f \\ \tau_f \end{vmatrix} \\ &\underline{\psi} = \begin{vmatrix} \frac{1}{2}u_f^{\,2} - \frac{1}{R_f} - C_E \\ R_f u_f \cos g_f - C_h \end{vmatrix} \end{aligned}\right\} (5.3.2)$$

However, as was the case in the example of section 2.5, J^* need not be considered to be a function of τ_f, since actual integration of the Euler-Lagrange equations provides a relationship between τ_f and the other final point variables. Hence $\underline{v}$ and $\underline{r}$ can be written more simply as

$$\underline{v} = R_f \qquad \underline{r} = \begin{vmatrix} u_f \\ g_f \\ R_f \end{vmatrix} . \tag{5.3.3}$$

Following the procedure described in section 2.3 yields

$$\frac{\partial \underline{\psi}}{\partial \underline{v}} = \begin{vmatrix} \dfrac{1}{R_f^{\,2}} \\ u_f \cos g_f \end{vmatrix} \tag{5.3.4}$$

$$\frac{\partial \underline{\psi}}{\partial \underline{w}} = \begin{bmatrix} u_f & 0 \\ R_f \cos g_f & - R_f u_f \sin g_f \end{bmatrix} \tag{5.3.5}$$

and finally

$$\Omega = \begin{vmatrix} \dfrac{1}{u_f R_f^{\,2}} \\ - \dfrac{\cos g_f (1 - u_f^{\,2} R_f)}{u_f^{\,2} R_f \sin g_f} \\ 1 \end{vmatrix} \tag{5.3.6}$$

Following the form of equation (2.3.10) the matrix $\left[\frac{\partial^2 J^*}{\partial \underline{r} \partial \underline{r}}\right]$ can be written as the sum of the following two matrices

$$\left[\frac{\partial^2 J^*}{\partial \underline{r} \partial \underline{r}}\right] = \left[\frac{\partial^2 G}{\partial \underline{r} \partial \underline{r}}\right] + \left[\frac{\partial \underline{\Lambda}_f}{\partial \underline{r}}\right] \qquad (5.3.7)$$

where $\underline{\Lambda}_f = (\lambda_{uf}, \lambda_{gf}, \lambda_{rf})$

The above form will always result for problems with fixed initial coordinates which can be written so that J* is not a function of t_f. The first term in equation (5.3.7) can be determined from equation (5.1.32) yielding,

$$\frac{\partial^2 G}{\partial \underline{r} \partial \underline{r}} = \begin{vmatrix} \mu_1 & - \mu_2 R_f \sin g_f & \mu_2 \cos g_f \\ - \mu_2 R_f \sin g_f & - \mu_2 R_f u_f \cos g_f & - \mu_2 u \sin g_f \\ \mu_2 \cos g_f & - \mu_2 u \sin g_f & - \frac{2\mu_1}{R^3} \end{vmatrix}$$

(5.3.8)

The variables μ_1 and μ_2 in the above expression are determined from equations (5.1.42) and (5.1.43) giving

$$\mu_1 = - R_f{}^2 \left[\frac{\lambda_{gf}}{R_f \tan g_f} + \lambda_{rf}\right] \qquad (5.3.9)$$

$$\mu_2 = \frac{\lambda_{gf}}{R_f u_f \sin g_f} \qquad (5.3.10)$$

The second term of equation (5.3.7) is determined by computational procedure explained in connection with equation (2.6.9). In this case M_f represents one of the elements of $\underline{\Lambda}_f$.

The computational procedure is as follows:

(1) The endpoint sufficiency test is initiated by a call for subroutine FOCAL. At this time matrix XOLD contains the time history of a trajectory to the endpoint which is to be tested.

(2) Subroutine FOCAL immediately calls subroutine CONSTRT. This subroutine used the nominal final point values $\underline{r}_f^*$ from XOLD to evaluate Ω from equation (5.3.6) and $\left[\frac{\partial^2 G}{\partial \underline{r} \partial \underline{r}}\right]$ from equations (5.3.8), (5.3.9) and (5.3.10). These values are returned to subroutine FOCAL.

(3) Subroutine FOCAL then evaluates the matrix $\frac{\partial \underline{\Lambda}_f}{\partial \underline{r}_f}$ in accordance with equation (2.6.9).

(4) Subroutine FOCAL then computes $\left[\frac{\partial^2 J^*}{\partial \underline{r} \partial \underline{r}}\right]$ using equation (5.3.7) and determines if it is positive definite. If the matrix is positive

definite, the endpoint being tested represents a local minimum.

To evaluate $\frac{\partial \underline{\Lambda}}{\partial \underline{r}_f}$ in step 3 requires the solution of three two-point boundary value problems. For example, to determine $\frac{\partial \underline{\Lambda}_f}{\partial u_f}$, a new final point is determined by making a small change Δ in u_f^*. An optimal path to this new endpoint having fixed final state coordinates must now be determined. The original boundary value problem involved the variable boundaries of equations (5.1.29) and (5.1.30) and the resulting transversality equations (5.1.47) and (5.1.48). For the new problem the final point constraints are

$$u_f = c_1 \tag{5.3.11}$$

$$g_f = c_2 \tag{5.3.12}$$

$$R_f = c_3 \tag{5.3.13}$$

With these constraints the only useful relation given by the transversality conditions is

$$H_f = 0 \tag{5.3.14}$$

Both the original two-point boundary value problem and the new problem can be solved using the same subroutine QUASI. This is due to the fact that QUASI does not directly determine the initial value updates from the boundary conditions. Instead, QUASI calls a separate subroutine to handle this job. Since this separate subroutine is an argument of subroutine QUASI, subroutine QUASI can be executed using any one of several different subroutines to handle differing sets of boundary conditions. The subroutine which handles the original variable final point problem is named BOUND. The subroutine which handles the fixed boundary values of equations (5.3.11) - (5.3.14) is named FIXED.

In the example, an optimal path to the new endpoint is determined by calling QUASI with subroutine FIXED as an argument and with $c_1 = u_f^* + \Delta$, $c_2 = g_f^*$ and $c_3 = R_f^*$. QUASI returns the new solution to FOCAL as a time history contained in matrix XOLD. Using the final values of the Lagrange multipliers from XOLD, the partial derivatives $\left[\frac{\partial \underline{\Lambda}_f}{\partial u_f}\right]$ can be computed directly using equation (2.6.9). The other two rows of $\left[\frac{\partial \underline{\Lambda}_f}{\partial \underline{r}_f}\right]$ are computed in similar fashion. Subroutines FOCAL, CONSTRT, FIXED and BOUND are listed in Appendix B.

5.3.2 The Overall Computational Scheme. The solutions to specific two-point boundary value problems exhibited in the next two sections were found following the general computational outline given below.

(1) Generate an initial solution estimate by integrating forward the nonlinear equations using the known initial values and estimates for unknown initial values. The purpose of this integration is to determine an estimate of the final time.

(2) With the final time known, it is possible to regenerate the initial solution by integration with fixed step length. The solution is simultaneously stored in the matrix XOLD at fixed equal increments in time, as required by subroutine QUASI.

(3) Solve the two-point boundary value problem using subroutine QUASI.

(4) If it is desired, determine if the resulting solution satisfies the endpoint sufficiency condition using subroutine FOCAL.

(5) Verify the solution determined by QUASI by integrating the nonlinear equations with the initial values found by subroutine QUASI.

(6) If other solutions are to be found to endpoints near the solution just determined, use the solution just determined as the initial solution estimate and return to step 3. If not, return to step 1 to solve a new problem.

Program SPACE, the first program in Appendix B, follows the above outline.

5.4 Examples of Multiple Stationary Solutions

In the course of the following discussions, reference will often be made to actual orbital transfers such as the one shown in Figure 5.4. The interpretation of these figures will be aided by the following conventions:

(1) The initial orbit is a circular orbit; initial motion in this orbit is counter-clockwise.

(2) Short lines eminating from the transfer trajectory at selected points as shown in Figure 5.4 indicate the true direction of the thrust vector. The tail of the thrust vector is always shown in contact with the transfer trajectory.

(3) The first thrust vector (proceeding in a counter-clockwise direction) on the initial orbit indicates the point of thrust initiation. The final thrust vector indicates the point of thrust termination.

(4) Although the thrust vector is displayed at a finite number of points along the transfer trajectory, the vehicle is thrusting continuously from the point of thrust initiation to the point

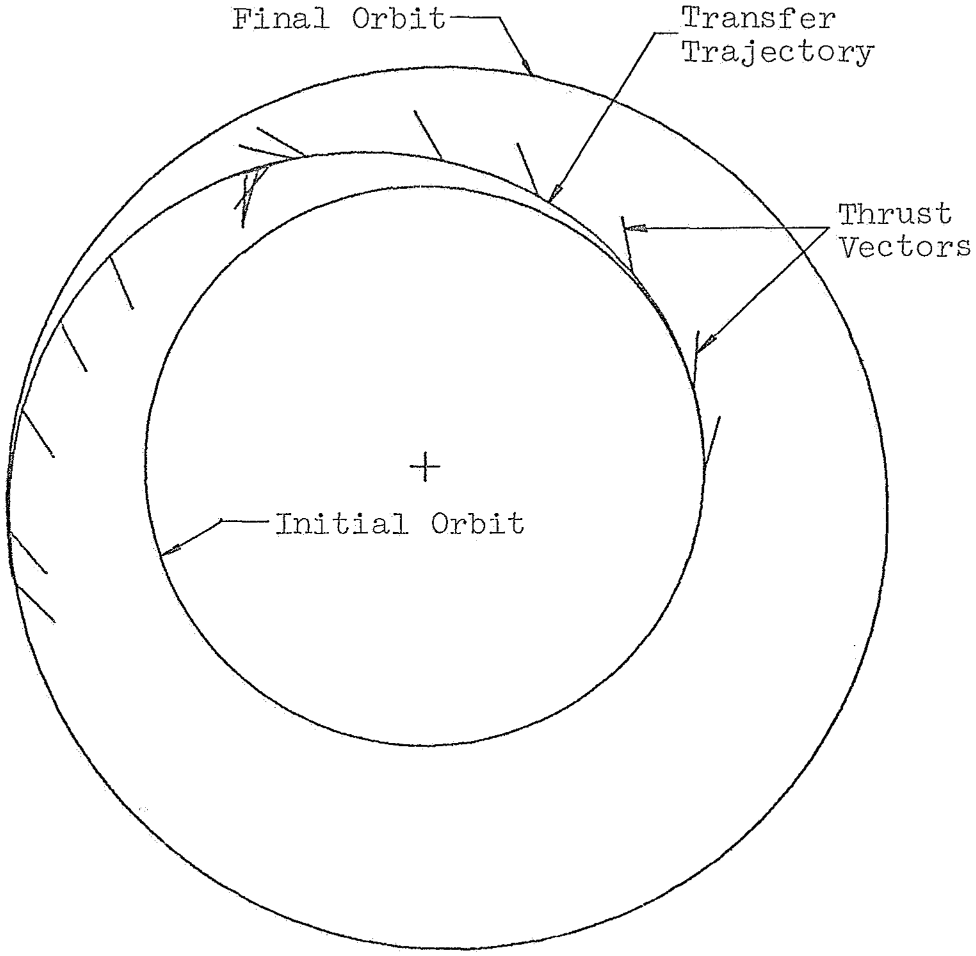

Fig. 5.4 Optimal Orbital Transfer to A = 1.58 and e = .11 (x_o = 16°)

of thrust termination. Likewise the thrust vector rotates in a continuous manner. The highly nonlinear nature of this rotation causes the orientation of the thrust vector to change suddenly at times as shown in Figure 5.4.

(5) Motion in the terminal orbit is in a counterclockwise direction, except where noted otherwise.

The computational procedure for implementing the endpoint sufficiency test, as described in the previous section, can be applied to the multiple stationary solutions discussed in section 5.2. From Figure 5.3 it can be seen that two solutions exist to point B. This point can be reached by starting integration with an initial control angle of either 4° or 16°.

Using the modified Newton-Raphson method for solving two-point boundary value problems (see Chapter 3), two solutions which terminated at a point near point B were found. Both solutions result in a final nondimensional semimajor axis A of 1.58 and a final eccentricity e of .110. The first trajectory begins with an initial angle of approximately 16° (15.5°). This orbital transfer is pictured in Figure 5.4. The thrust vector control angle increases from 16° to over 50° and then suddenly decreases rapidly to -76°. The control angle then slowly increases to -36°. In this case the state space endpoint is given by $u_f = .826$, $g_f = 5.92^\circ$, and $R_f = 1.52$. It is readily verified that these states correspond to $A_f = 1.58$ and $e_f = .110$. Using the computational procedure discussed in section 5.3, the above endpoints were found to satisfy the endpoint sufficiency test.

The second trajectory initiated transfer with $x = 4^\circ$ (4.4°). This orbital transfer is shown in Figure 5.5. In this case the thrust-vector control angle increases slowly to approximately 80° and then suddenly swings past 180° to approximately -100°. The control angle then increases at a slower rate until $x_f = -31^\circ$. Note that the thrust vector has actually been in a "retro-fire" position for a short period of time during this maneuver. The trajectory terminates with the following final status: $u_f = .773$, $g_f = 6.10^\circ$, and $R_f = 1.62$. These states, as before, correspond to $A_f = 1.58$ and $e_f = .110$. Numerical results show that this trajectory and its resultant endpoints do not satisfy the endpoint sufficiency conditions. The corresponding trajectory is therefore nonoptimal. In fact, the above nominal endpoints represent a local maximum value for the functional J*.

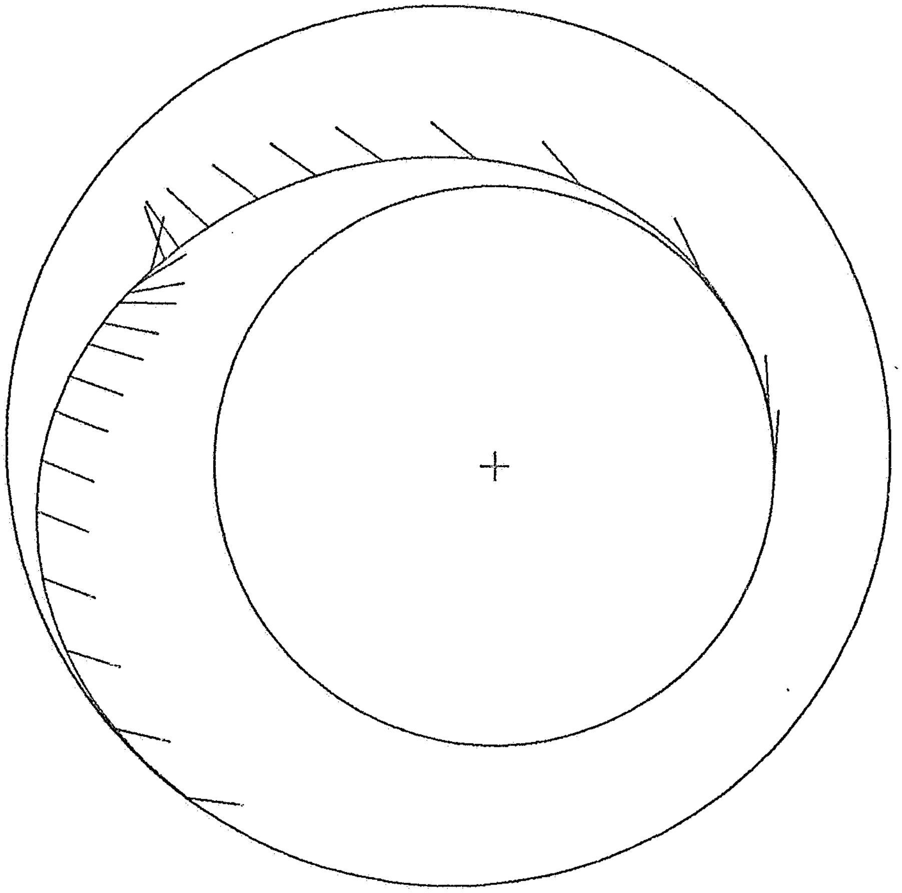

Fig. 5.5 Nonoptimal Orbital Transfer to A = 1.58 and e = .11 (x_o = 4°)

This fact was substantiated by computing neighboring stationary trajectories to fixed final points in the vicinity of the nominal final points. These final points were chosen beforehand to correspond to $A_f = 1.58$ and $e_f = .110$. In all cases these neighboring trajectories had performance indices less than those of the nominal trajectory.

Since $T = T_{max}$ from $\tau = 0$ until $\tau = \tau_f$, the terminal time is proportional to the magnitude of the performance index J of equation (5.1.25). The value of τ_f is therefore a convenient measure of the performance of a given maneuver. For the nonoptimal transfer initiated with $x_o = 4^o$, the value of τ_f was found to be 5.89. For the optimal transfer initiated with $x_o = 16^o$, the value of τ_f was 4.38. The optimal trajectory therefore yields a performance index which is 25% less than that for the nonoptimal trajectory.

Figure 5.6 shows the trajectories in A-e space corresponding to the optimal and nonoptimal orbital transfers shown in Figures 5.4 and 5.5, respectively. A trajectory in A-e space should not be confused with the locus of optimal endpoints previously discussed. A trajectory in A-e space depicts the values of A and e which were attained during the progress of the transfer.

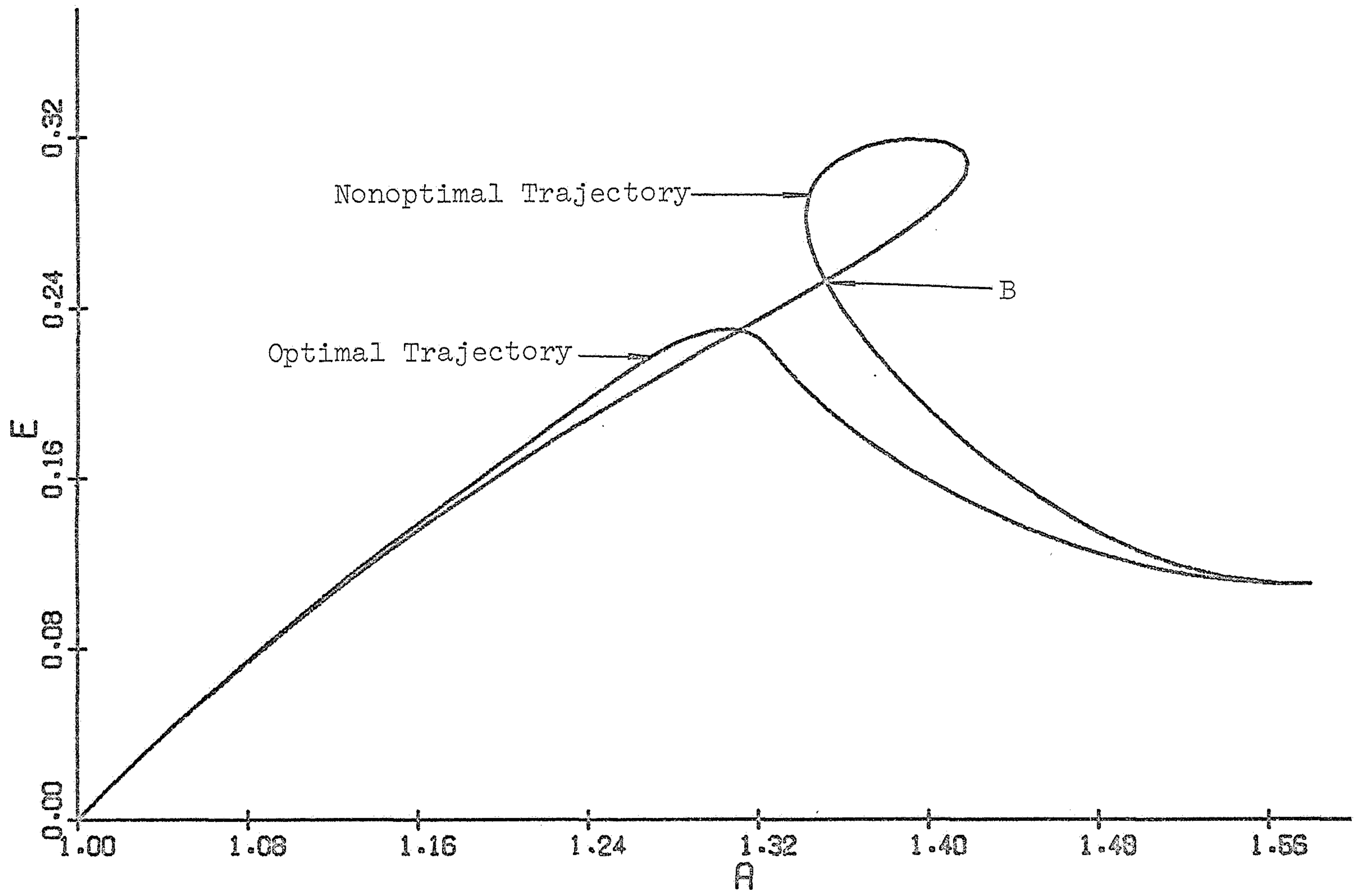

Fig. 5.6 An Optimal and a Nonoptimal Trajectory to A = 1.58 and e = .11

The loop in the nonoptimal trajectory corresponds to the sudden rotation of thrust vector to the retro-fire position as depicted in Figure 5.5.

If the thrust magnitude were an allowable control, this transfer would obviously be nonoptimal since the thrust could then be turned off at point B in Figure 5.5 and reignited at a later time. Doing so would save all the effort needlessly expended traversing the loop. It is interesting to note that the argument of periapsis for the nonoptimal final orbit (337°) is almost diametrically opposite to that of optimal final orbit (119°), where periapsis is measured in a counter-clockwise direction from the point of thrust initiation.

In terms of the locus of "optimal" endpoints shown in Figure 5.2, one endpoint on branch CD has been determined to be nonoptimal. The application of the endpoint sufficiency test to other endpoints on branch CD has shown these endpoints to be nonoptimal, whereas, other endpoints on endpoint locus OABC have consistently passed the endpoint sufficiency test. The test is indeterminate at point C, since this point represents a final circular orbit and $\sin g_f = 1 - u_f^2 R_f = 0$ in equation (5.3.6).

Since similar results have been observed for loci generated with initial control angles other than 4°, the above conclusions can be generalized to eliminate from consideration branches corresponding to branch CD. This generalization applies only if the locus of optimal endpoints exhibits the same qualitative characteristics as those shown in Figure 5.2. With such branches eliminated from consideration, the multiplicity of solutions shown in Figure 5.3 disappears.

A second example of multiple stationary solutions is illustrated in Figures 5.7 - 5.9. In this case the optimal trajectory shown in Figure 5.7 starts with $x_o = 3.53^o$ and has a thrusting period of 6.10 nondimensional time units. The nonoptimal trajectory shown in Figure 5.8 initiates thrusting with $x_o = - 4.26^o$ and has a thrusting period of 7.50 nondimensional time units. The latter trajectory was numerically determined to be nonoptimal through the use of the endpoint sufficiency condition. In this case the optimal trajectory has a performance index which is more than 20% less than that for nonoptimal transfer. The qualitative remarks pertaining to the first example also apply in this case. The main difference between these two examples is a computational one. Both trajectories in the first example were terminated when the first zero of the cutoff function was encountered. However, the nonoptimal trajectory

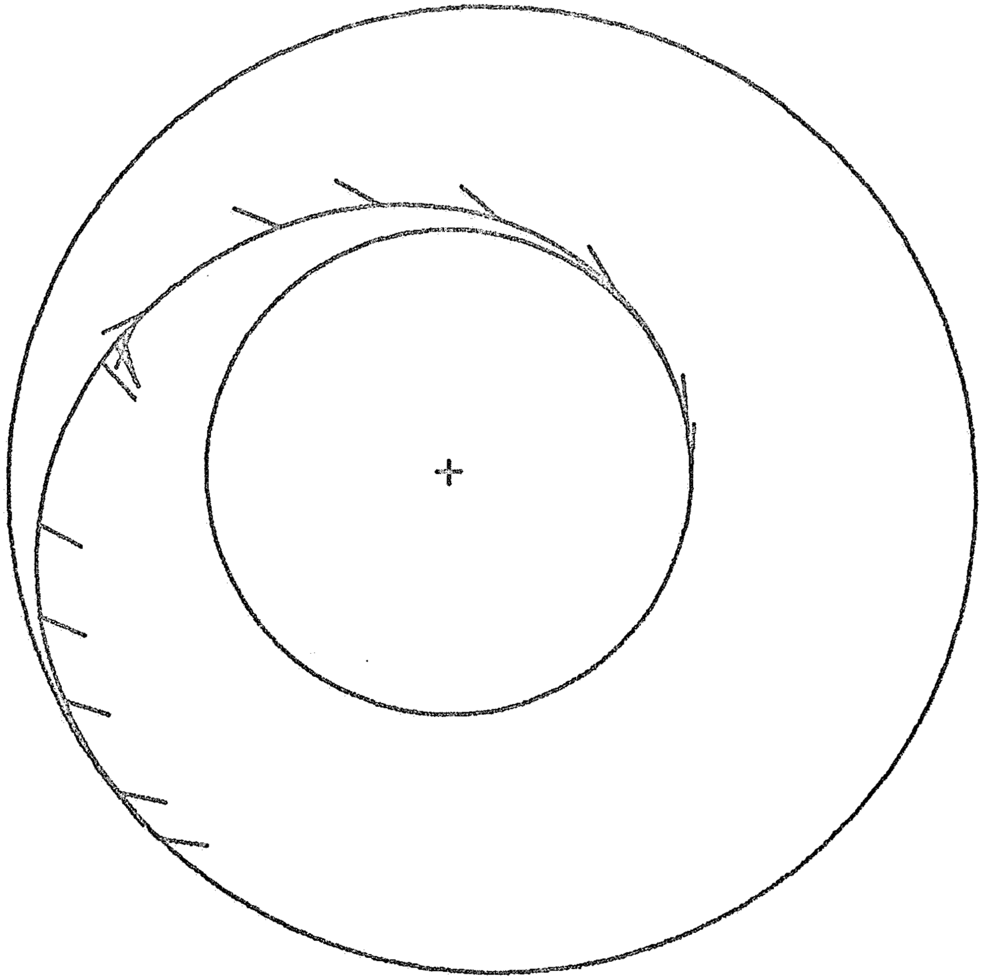

Fig. 5.7 Optimal Orbital Transfer to A = 2.00 and e = .10 (x_o = 3.53°)

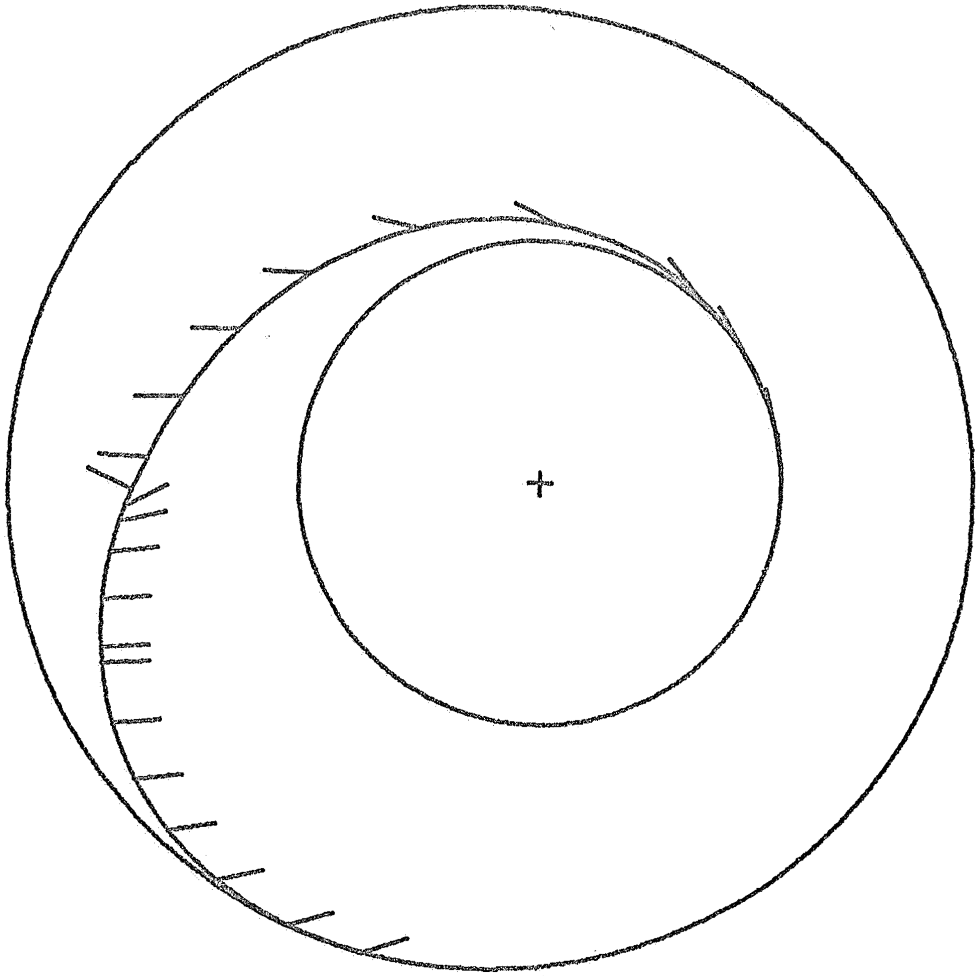

Fig. 5.8 Nonoptimal Orbital Transfer to $A = 2.00$ and $e = .10$ ($x_o = -4.26$)

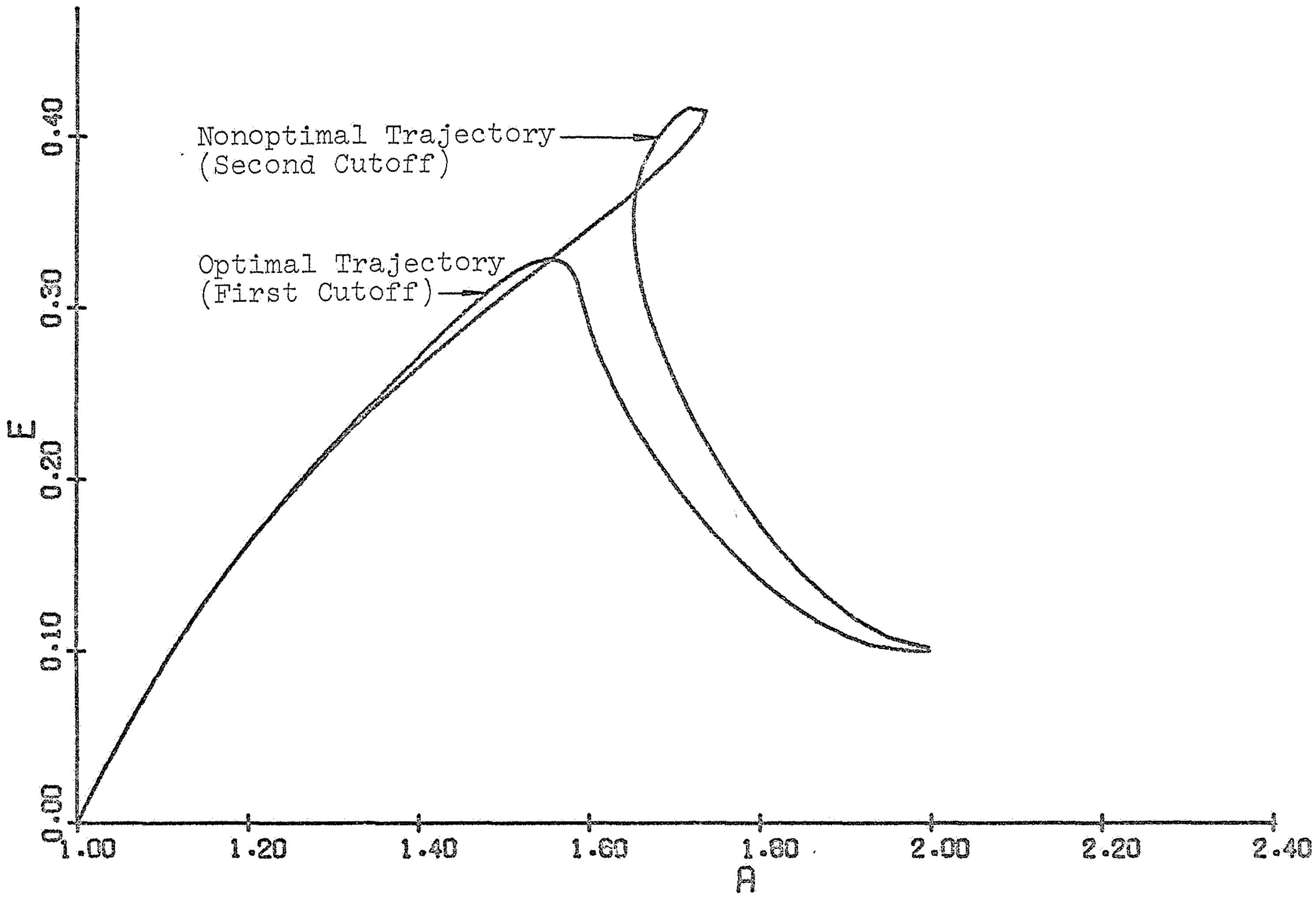

Fig. 5.9 An Optimal and a Nonoptimal Trajectory to A = 2.0 and e = .11

of the second example shown in Figure 5.8 was terminated when the second zero of the cutoff function was encountered.

The endpoint sufficiency test is an effective computational tool because it allows the rejection of certain nonoptimal solutions. When multiple solutions are encountered,the true optimal is easily distinguished via a direct comparison of the performance indices of the two solutions. Using the endpoint sufficiency test, a complete class of nonoptimal solutions can be discarded immediately upon encounter. Without the aid of the endpoint sufficiency test an investigator has no indication that solutions he is generating are nonoptimal until he encounters multiple solutions. Hence, an overall savings in computation time is realized by using the endpoint sufficiency test since the amount of time wasted generating nonoptimal solutions is greatly reduced.

As shown in section 1.3, multiple solutions can also arise if path sufficiency conditions are not met. As an example, Figures 5.10 - 5.12 summarize a case in which two orbital transfers exist to A = .835 and e = .270, both of which satisfy the endpoint sufficiency condition. The optimal transfer initiates thrusting with $x_o = 47^\circ$, but eventually finishes the transfer with a period in which the thrust vector is in a retro-firing

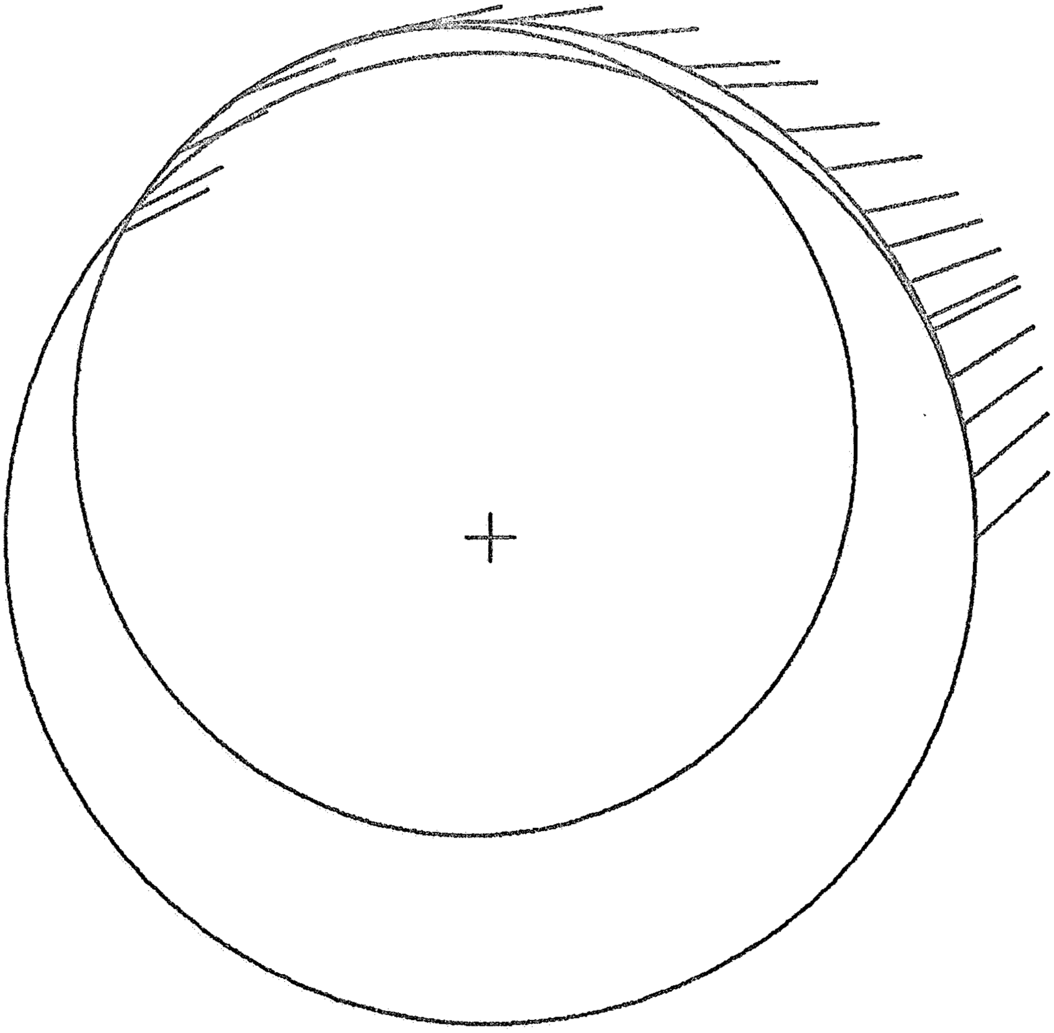

Fig. 5.10 An Optimal Orbital Transfer to A = .835
and e = .270

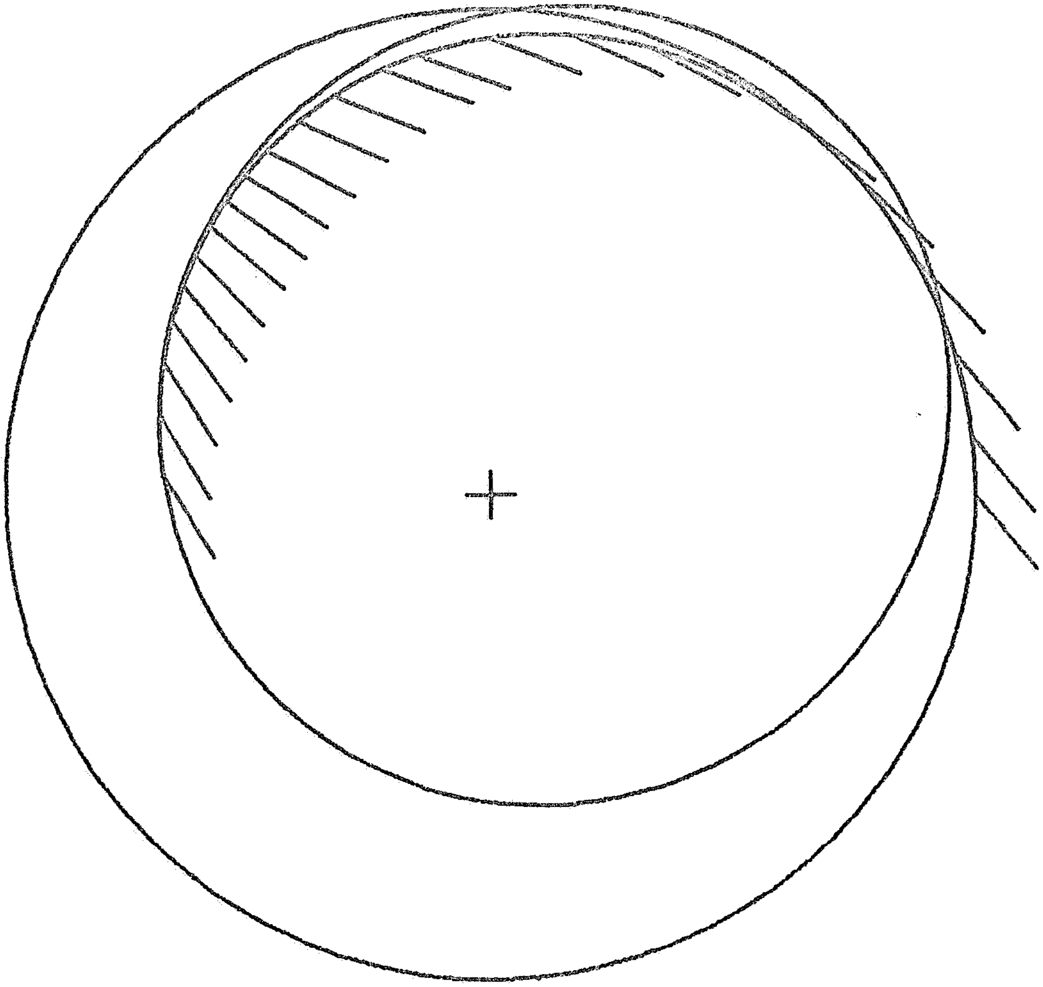

Fig. 5.11 A Nonoptimal Orbital Transfer to A = .835 and e = .270 which Satisfies Endpoint Sufficiency Condition

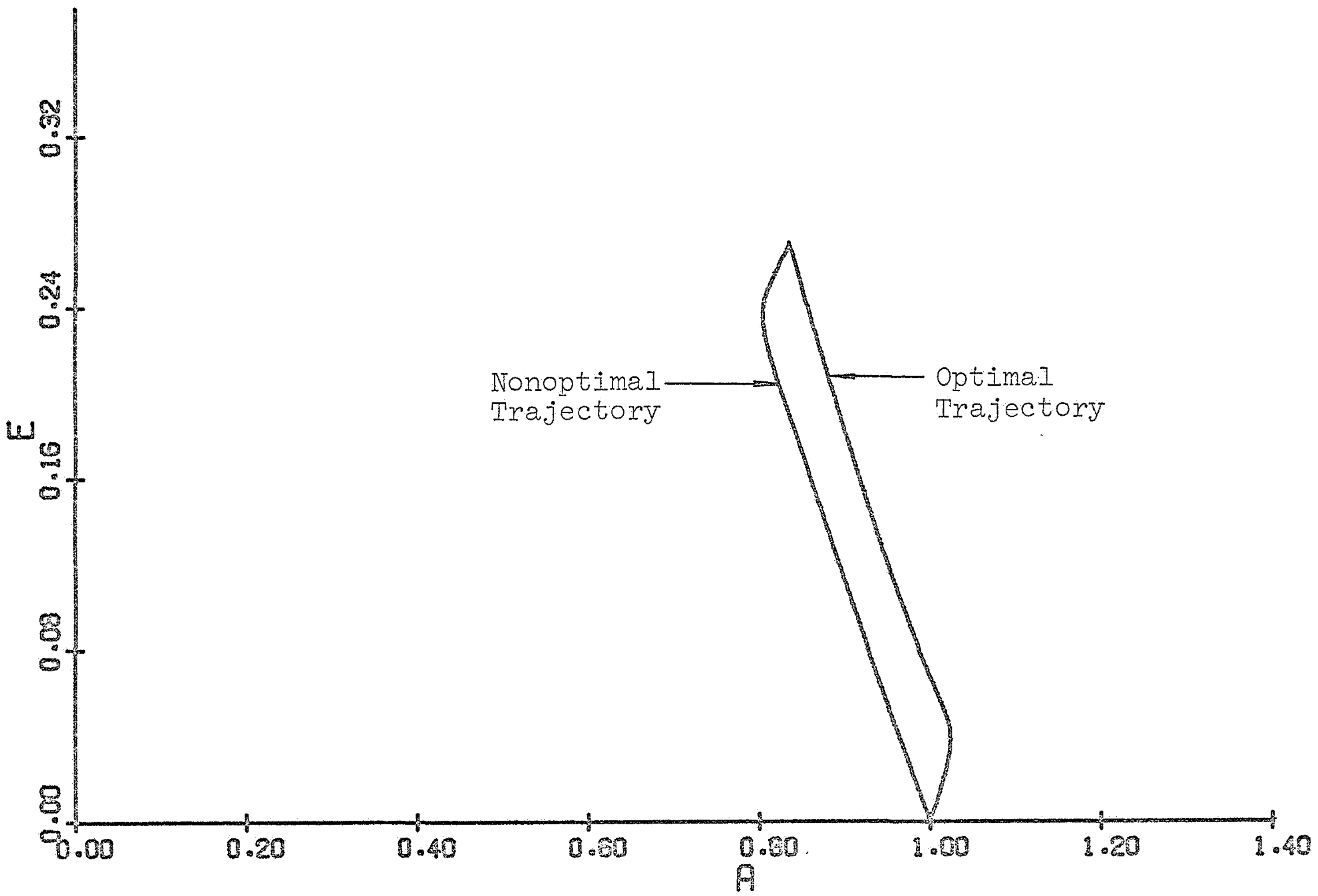

Fig. 5.12 Two Orbital Trajectories to A = .835 and e = .270 which both Satisfy the Endpoint Sufficiency Condition

position. The nonoptimal transfer is initiated with a retro-firing period followed by a period with a component of the thrust in the forward direction. The antisymmetry of these two transfers is apparent from Figure 5.12. In this case the optimal trajectory has been distinguished from the nonoptimal trajectory by a direct comparison of performance indices. The optimal transfer is achieved with a thrust duration of 2.71 nondimensional time units, while the nonoptimal transfer requires 2.75 time units. Future investigations can explore the possibility of eliminating multiple solutions of this type in a direct manner by implementing a path sufficiency test.

5.5 Optimal Orbital Transfer Results

This section presents further results concerning the minimum fuel orbital transfer problem described in section 5.1. The investigation was again restricted to problems in which the thrust magnitude was constant. In addition only a single set of space vehicle parameters were investigated. A nondimensional thrust F = .06666667 and an exhaust velocity $u_e = 10^{+50}$ were used. The thrust was chosen to coincide with that used by Mason (1967). Rather than choose a single arbitrary value for the exhaust velocity, it was set equal to the limiting value of infinity. Using the modified Newton-Raphson method to solve the two-point boundary value problem (see Chapter

3) and the endpoint sufficiency test to aid in the determination of the true optimality of solutions, optimal trajectories were computed to orbits having a wide range of final semi-major axis and eccentricities.

Figure 5.13 shows the loci of optimal endpoints determined for various initial thrust-vector control angles x_o. There are no intersections among these loci. As mentioned earlier, various regions of this endpoint space are attained by terminating the thrust at various zeros of the cutoff function. Figure 5.14 depicts regions in which the first, second, third, and fourth zero have been used. Crosshatched areas indicate regions of uncertainty. Final orbits represented by points in the region labeled "1" are attained by terminating thrust when the first zero of the cutoff function is encountered. Final orbits represented by points in regions labeled "2" are attained by terminating thrusting after the second zero of the cutoff function has been encountered. Similar statements can be made concerning the regions labeled "3" and "4."

Some of these boundaries seem to have little physical significance. For example, a comparison between trajectories in the regions labeled A and those in the region labeled B shows there to be no sudden discontinuity of any kind across the boundary. The same is true

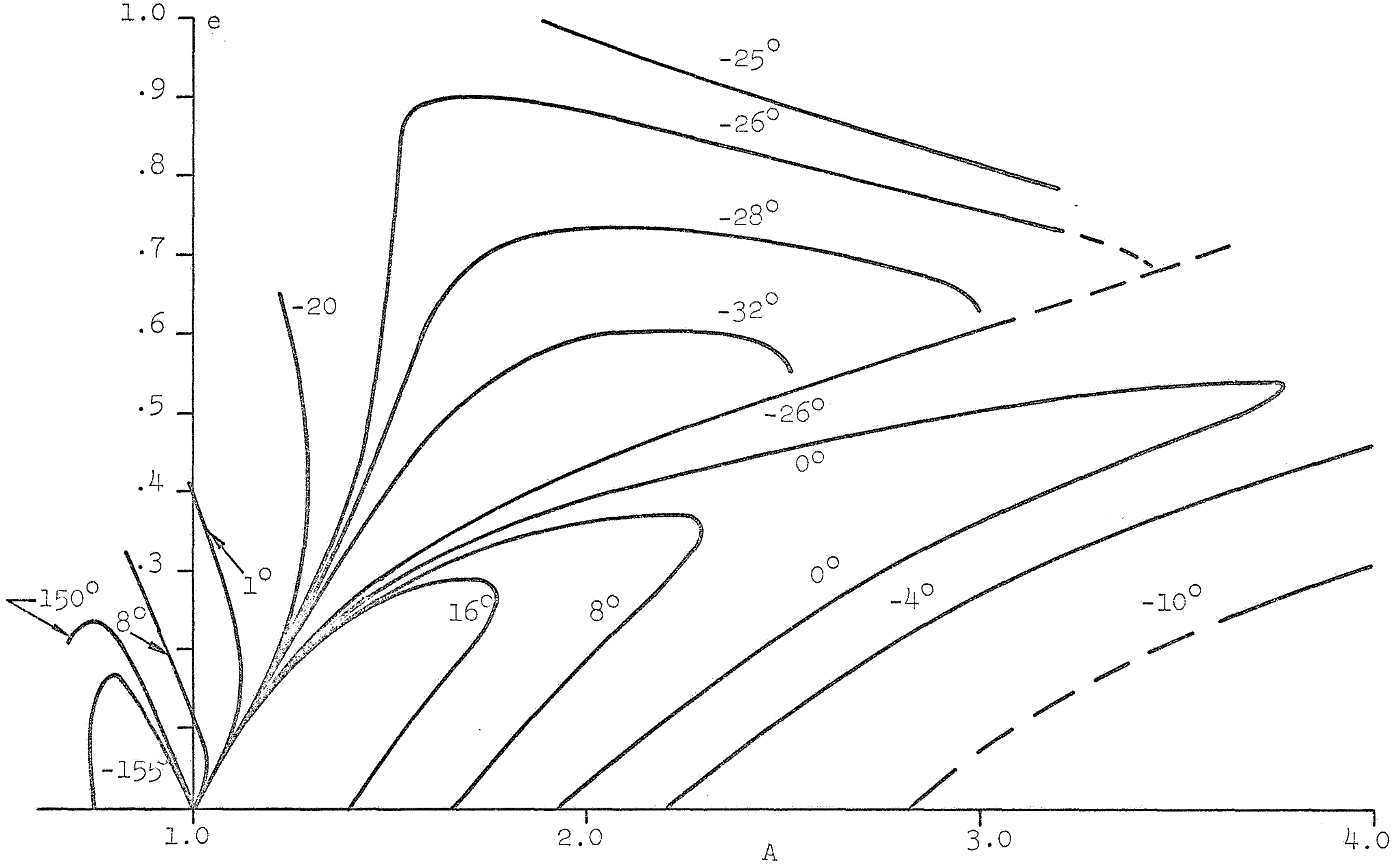

Fig. 5.13 Loci of Optimal Endpoints for Various Initial x_o

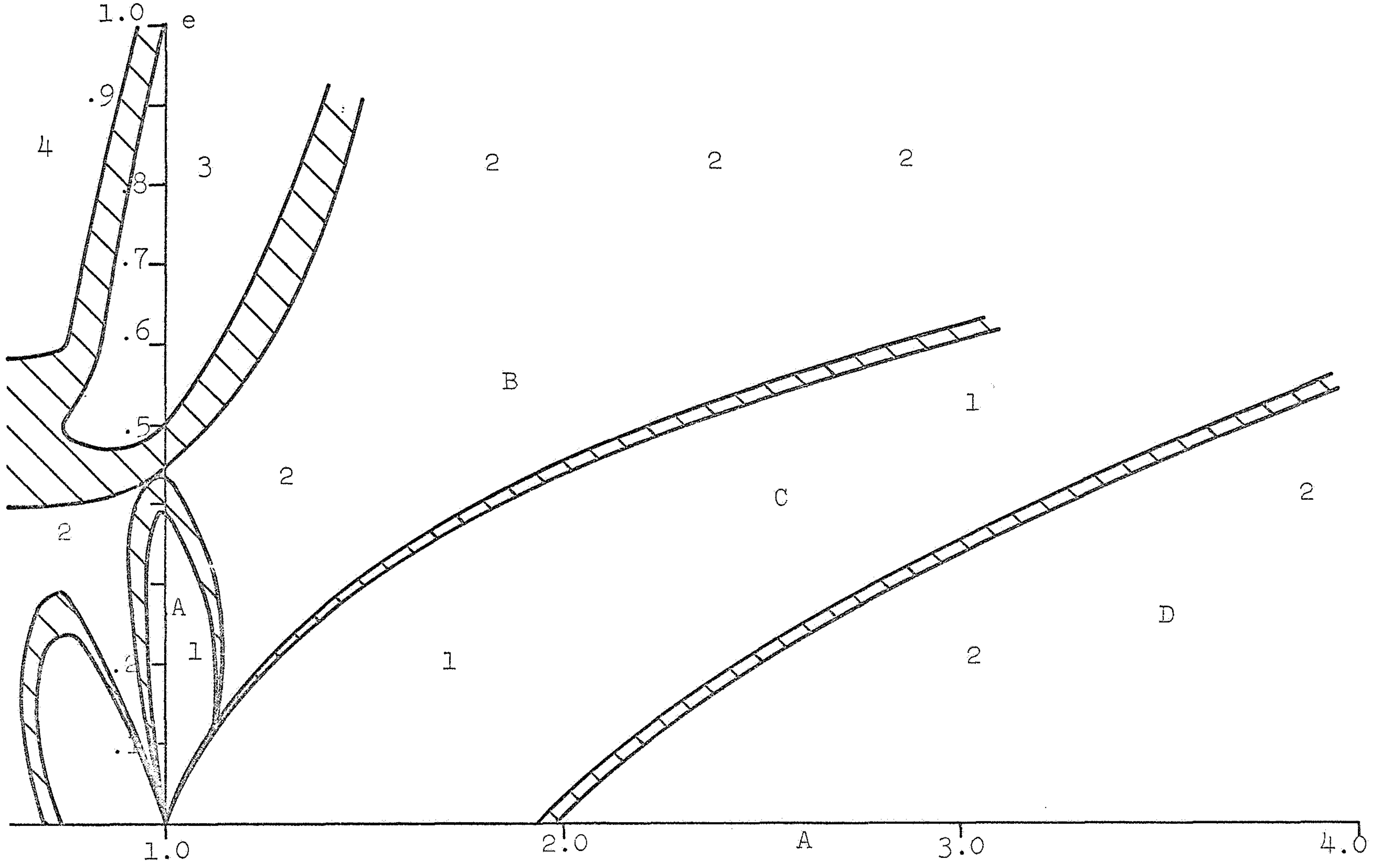

Fig. 5.14 Optimal Number of Cutoffs to Attain Various Orbits

for the boundary between regions C and D. This is not true, however, for the boundary between regions B and C. This fact is apparent from Figure 5.13. In this figure the boundary between regions B and C lies just above the line with positive slope labeled -26°. From Figure 5.13 it is seen that there is a jump in initial angle x_o across this boundary, viz., -32° to -26°.

An explanation for this jump is shown in Figure 5.15. Curve A shows the cutoff function for a trajectory initiated with $x_o = -32^o$ and $x_o = .46$. Thrusting is terminated at the second zero of this function. This corresponds to an initial orbital transfer to a point at the end of the curve labeled -32 in Figure 5.13. Curve B in Figure 5.15 shows the cutoff function for a trajectory initiated with $x_o = -32^o$ and $\dot{x}_o = .44$. Although only a slight change has been made in the cutoff function, the second cutoff does not occur, and the corresponding orbital transfer is to a hyperbolic orbit. This accounts for the physical discontinuity of orbital transfers to points just on either side of the boundary.

Figure 5.16 depicts the transfer time needed to attain an entire range of various final orbits. The transfer time is proportional to the performance index of equation (5.1.25). To attain a given final semimajor axis, this figure shows it is easier to reach an

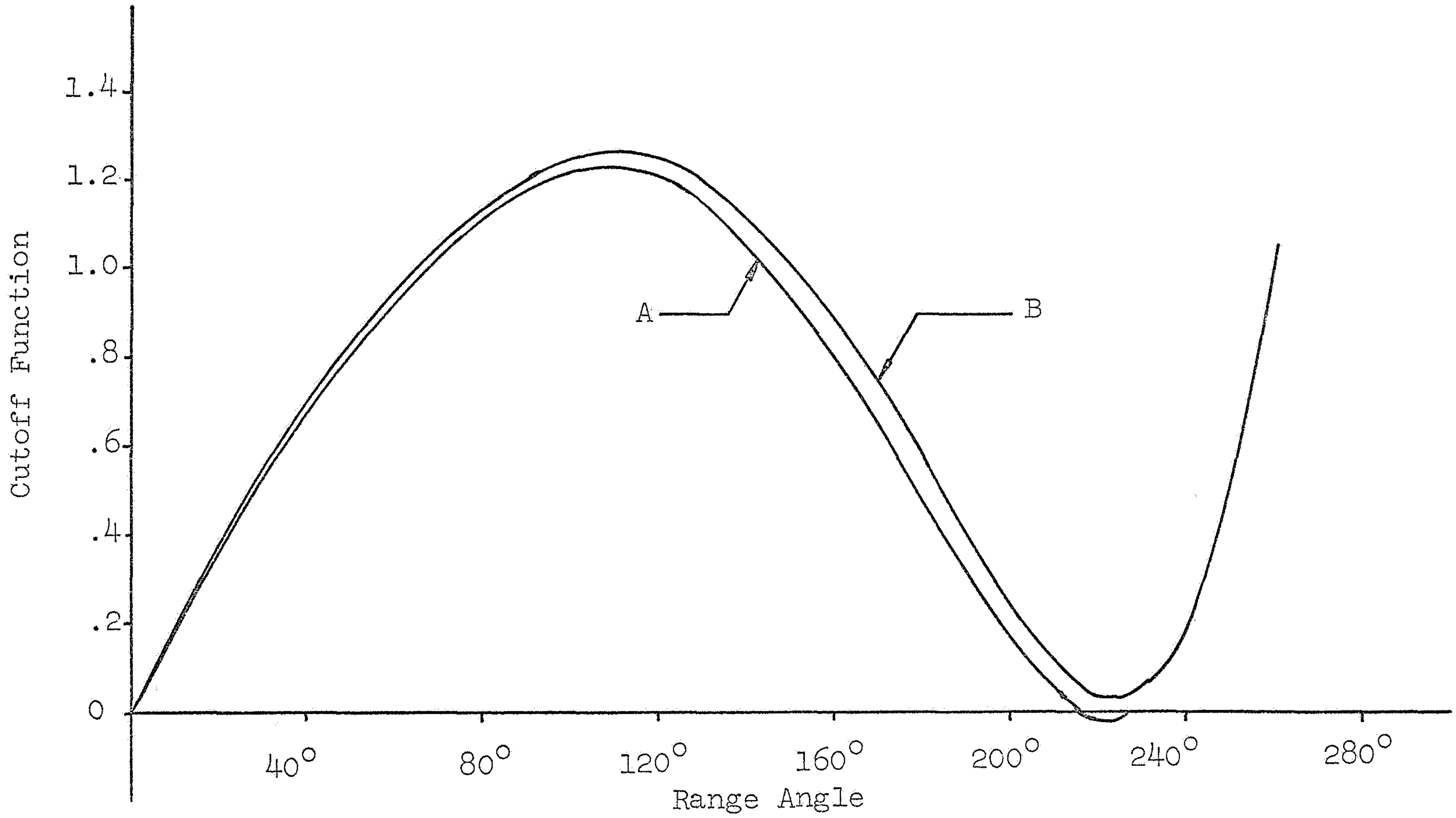

Fig. 5.15 Cutoff Function for Two Neighboring Trajectories

orbit with an intermediate eccentricity than it is to reach a circular orbit or an orbit with high eccentricity. The dashed line in Figure 5.16 represents the locus of final eccentricities associated with minimum effort transfers to any given semi-major axis provided no constraint is placed on eccentricity.

Two fundamental principles of astrodynamics can be verified by observing Figure 5.17. This figure pictures an orbital transfer to a final orbit with a semi-major axis of 1.5 and an eccentricity of .60. This is a typical orbital transfer requiring a substantial change in both energy and angular momentum. From the figure it can be seen that the thrust vector is essentially aligned with the velocity vector during the portion of transfer nearest the center of attraction. It is well known that a small change in the vehicle's kinetic energy is given by

$$\Delta E = m\underline{v}\cdot\Delta\underline{v} \qquad (5.5.1)$$

Therefore, the energy of the vehicle is increased in the most economic fashion by thrusting in the direction of the vehicle velocity when the vehicle velocity is a maximum. For low thrust orbits this generally occurs at points on the transfer trajectory nearest to the center attraction. In Figure 5.17 the space vehicle is using

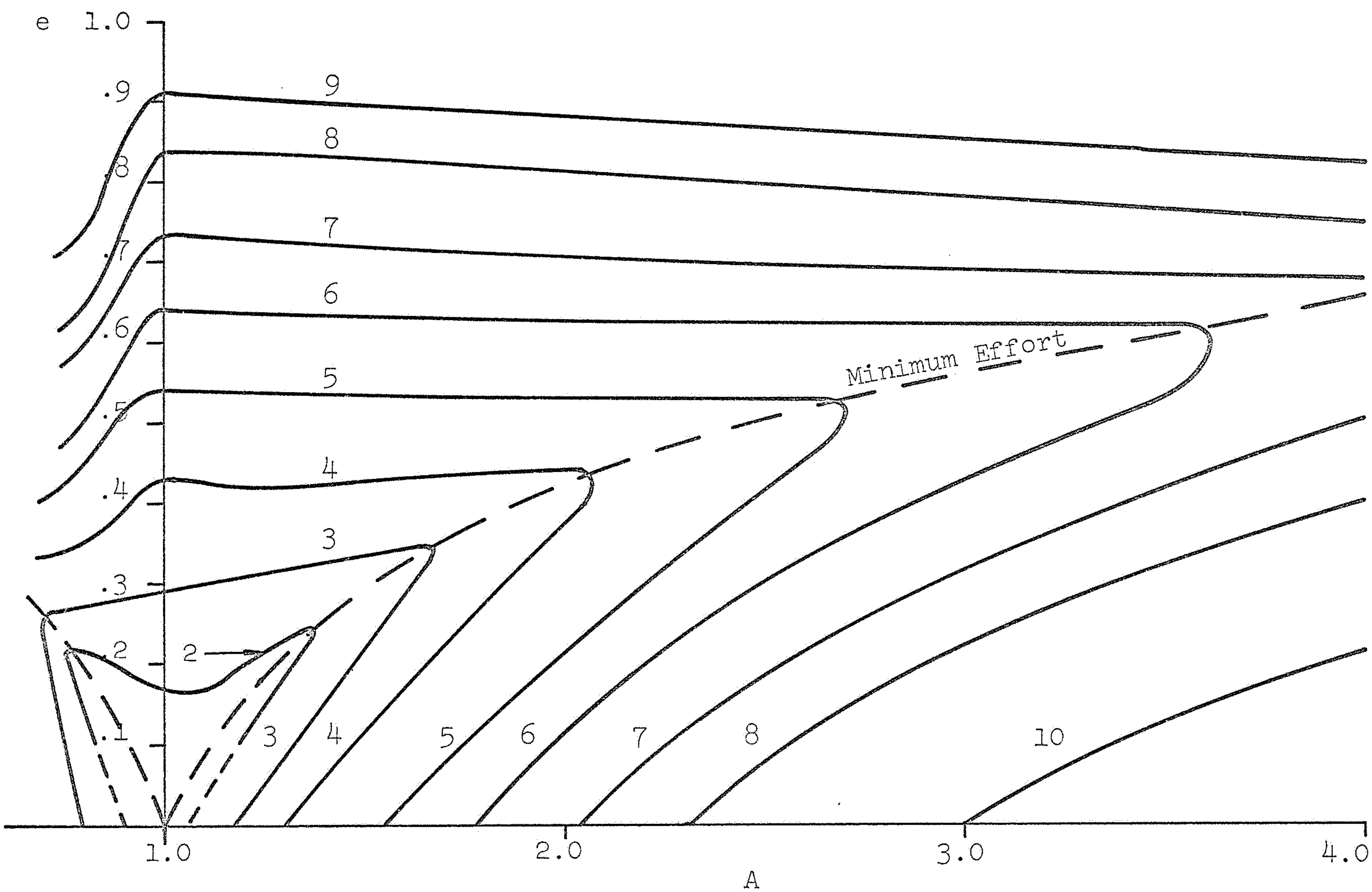

Fig. 5.16 Optimal Orbital Transfer Times

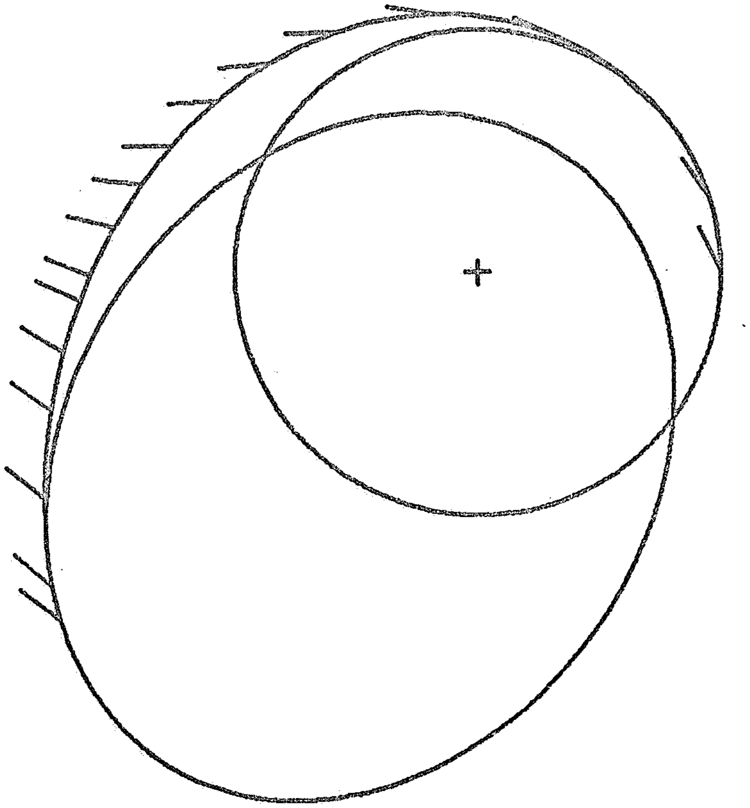

Fig. 5.17 A Typical Optimal Orbital Transfer and Fundamental Principles of Astrodynamics

its thrusting capability to best advantage by increasing its energy during the initial portion of the orbit.

On the other hand a change in angular momentum is most economically accomplished by thrust perpendicular to the radius vector at the local apogee of the transfer. This is easily seen from the following equation

$$\Delta\underline{h} = m\underline{r} \times \Delta\underline{v}$$

The product $\underline{r} \times \Delta\underline{v}$ is maximized by thrusting in a direction perpendicular to the radius vector at points along the transfer trajectory at which r is a maximum.

It will be seen later that an optimal maneuver for making large changes in eccentricity, but not in energy, requires that the vehicle first make a large gain in energy. Then at large radii the eccentricity is economically changed. Figure 5.18 shows an orbital transfer yielding an extreme change in angular momentum. In fact, the sense of the angular momentum vector reverses during the transfer and orbital motion changes from counter-clockwise to clockwise.

Figures 5.19 - 5.21 illustrate several optimal orbital transfers to final circular orbits. The transfer shown in Figure 5.19 is typical of optimal transfers to final circular orbits with $1.0 > R > 1.98$. These transfers are characterized by a sudden rotation of the

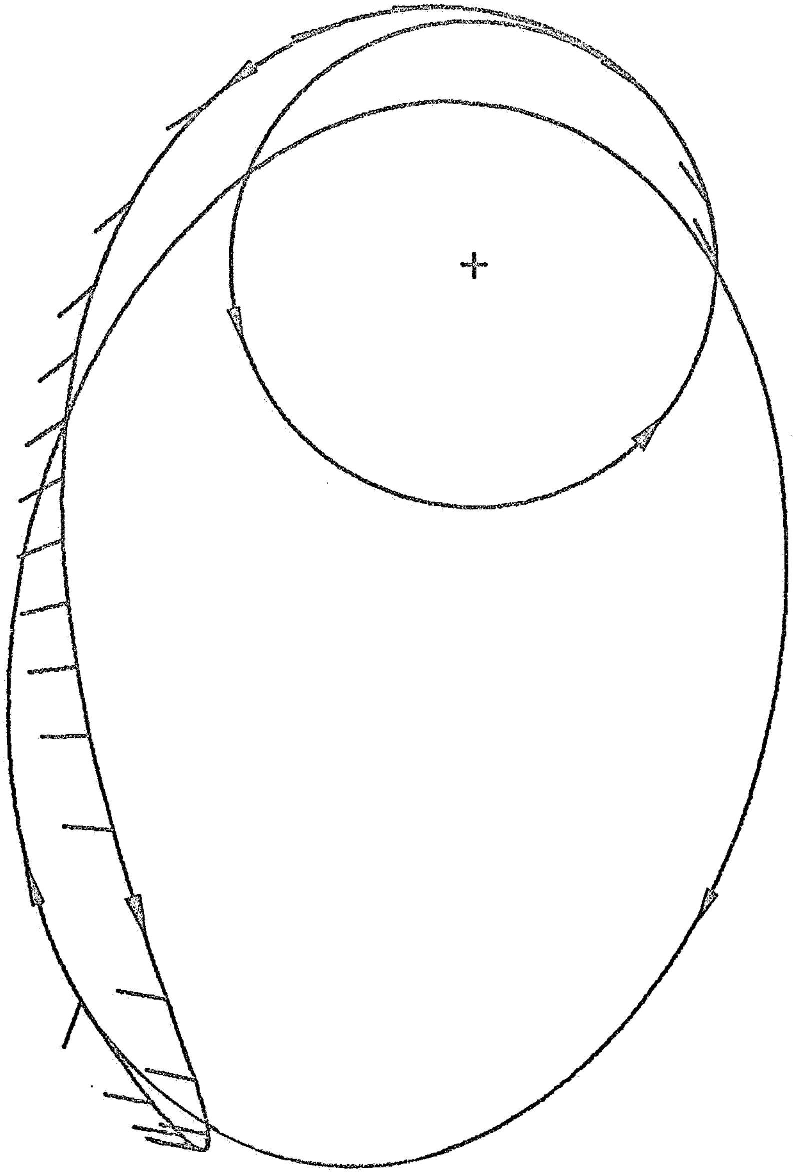

Fig. 5.18 An Optimal Orbital Transfer which Changes the Sense of the Angular Momentum Vector

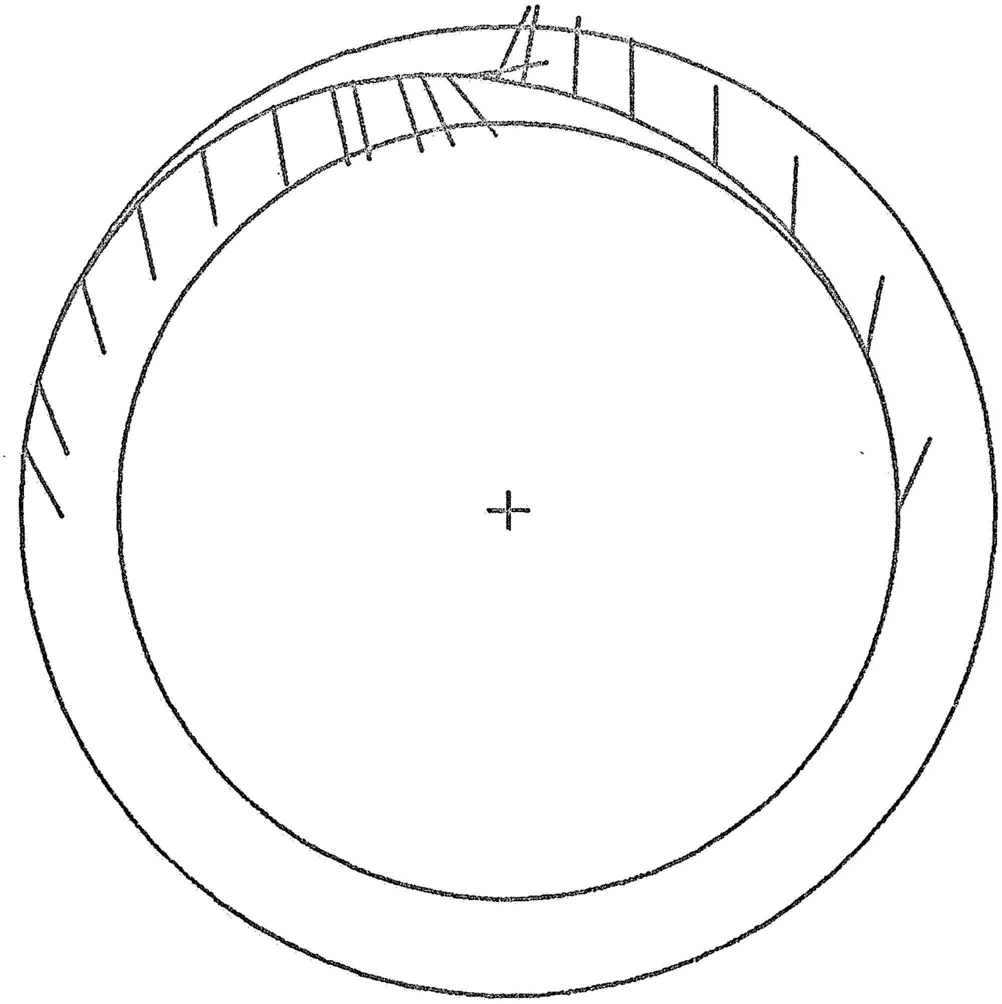

Fig. 5.19 An Optimal Orbital Transfer to a Circular Orbit of Radius 1.25

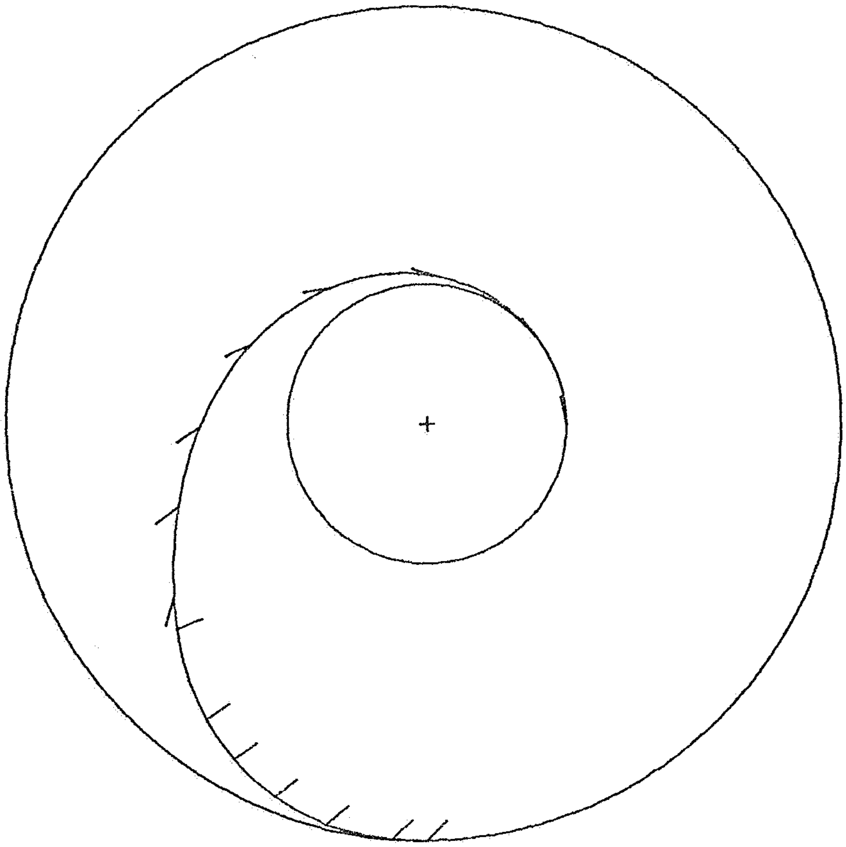

Fig. 5.20 An Optimal Orbital Transfer to a Circular Orbit of Radius 3.00

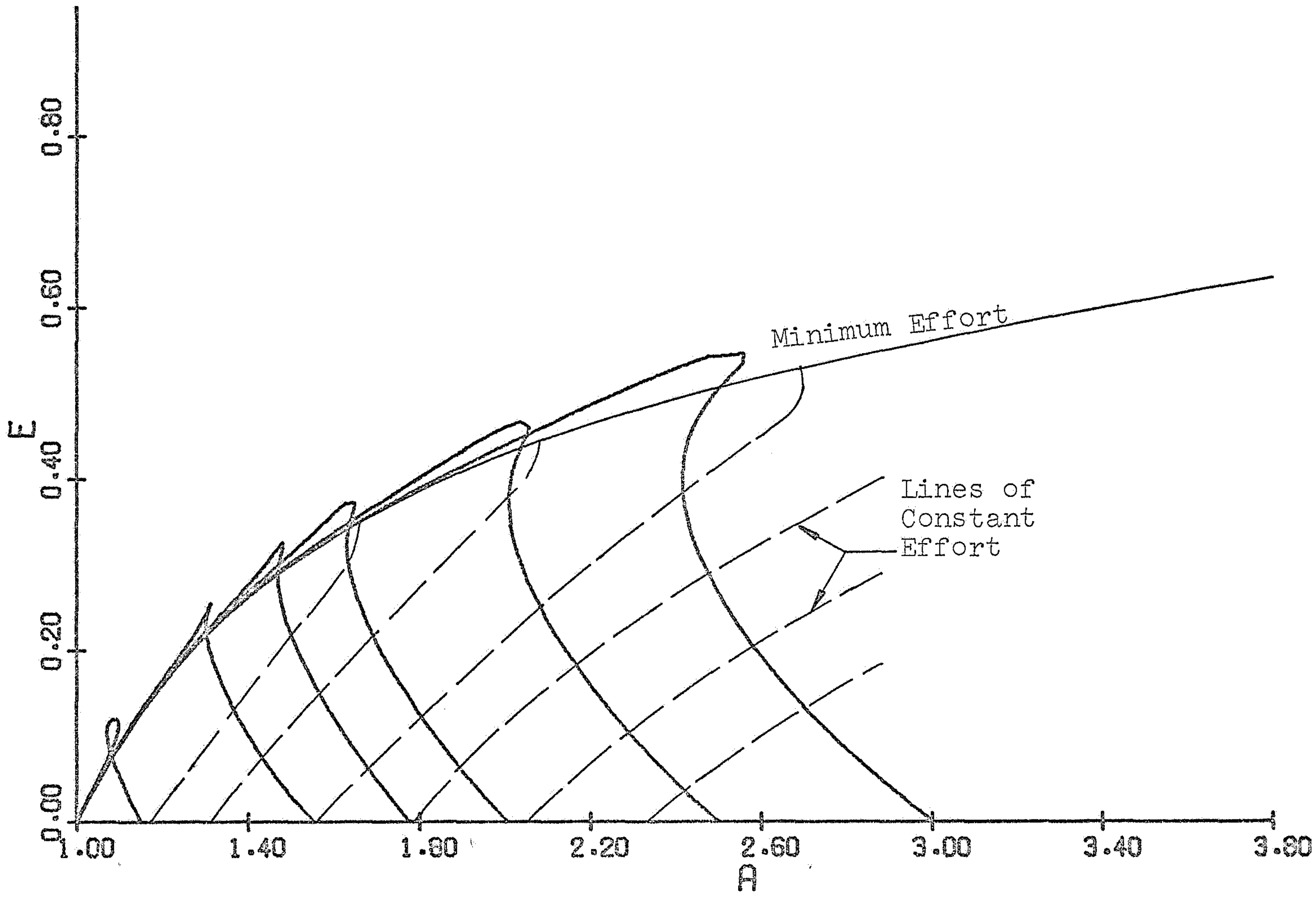

Fig. 5.21 Several Optimal Trajectories to Circular Final Orbits

thrust vector through a retro-fire position. This often generates loops in the A-e space trajectories (see Figure 5.21) which have previously been associated with nonoptimal transfers. As previously mentioned, the end-point sufficiency condition is indeterminate for transfers to circular orbits. Loops in A-e trajectories may be a result of requiring the thrust magnitude to be constant. For near orbit transfers a Homan impulsive transfer (Lawden, 1963, pp. 106-110) proves to be more optimal and one may conclude that thrusting at intermediate radii is to be avoided under certain circumstances. In a sense the loop represents a period in which the thrust should be off.

The transfer shown in Figure 5.20 is typical of optimal transfers to circular orbits with $R > 2.00$. In these cases the thrust vector exhibits a highly nonlinear oscillation about $x = 0$, but never passes through the retro-fire position. In Figure 5.21 notice that trajectories to circular orbits first pick up a substantial portion of the final energy required by proceeding along the path of minimum effort. The eccentricity is then adjusted to zero by proceeding almost perpendicular to lines of constant effort.

Figures 5.20, 5.22 and 5.23 summarize transfers to final orbits with a semi-major axis of 3.0. The

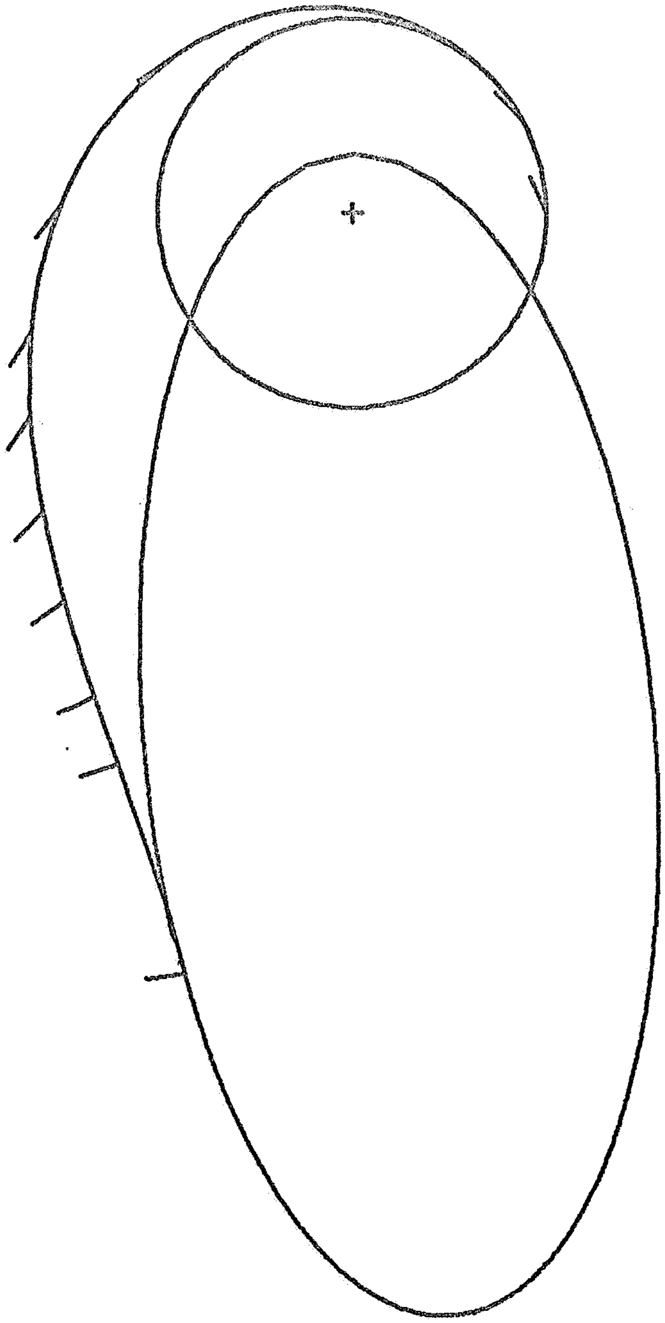

Fig. 5.22 An Optimal Orbital Transfer to A = 3.0 and e = .90

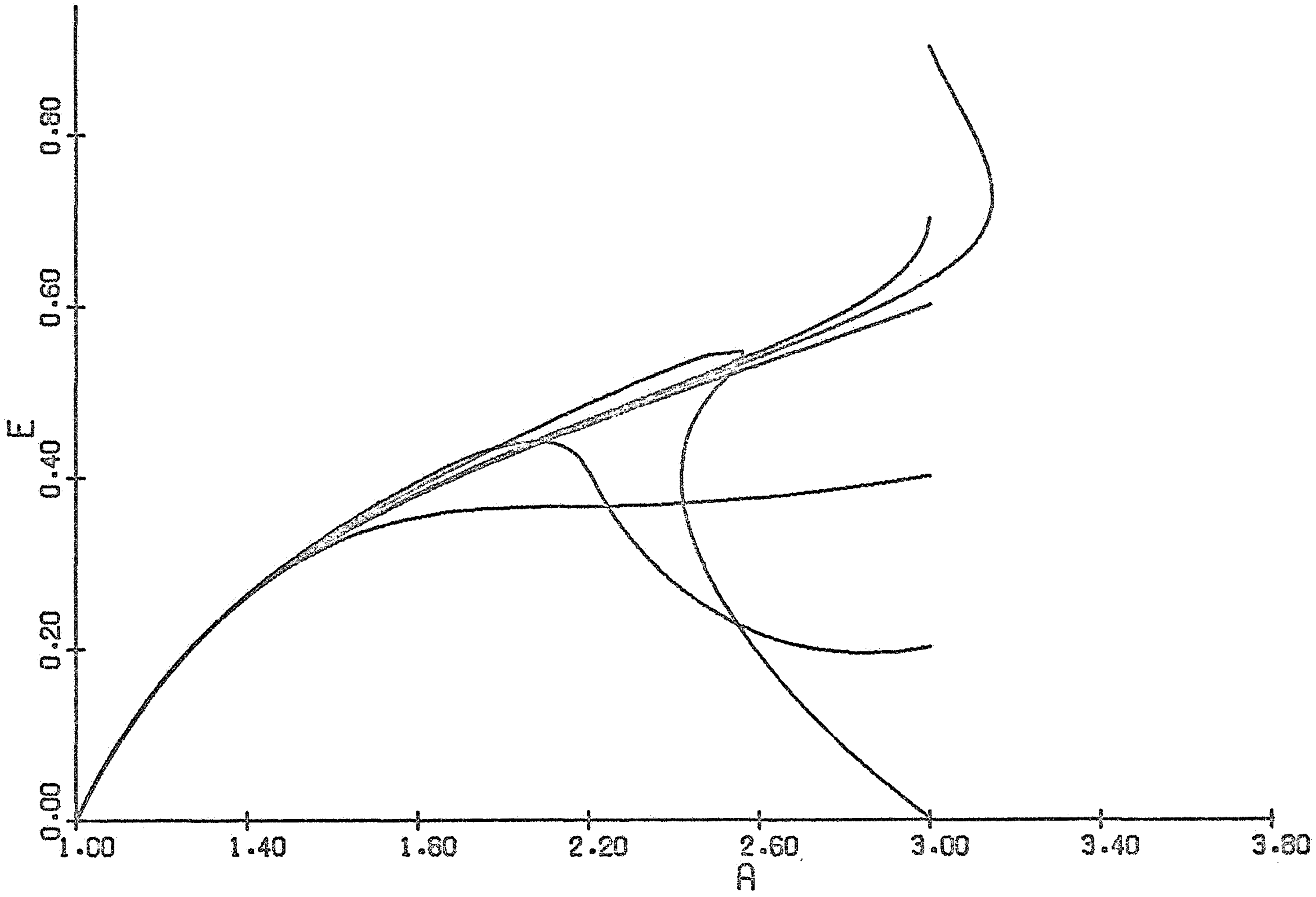

Fig. 5.23 Several Optimal Trajectories to Final Orbits with A = 3.0

fundamental principles of astrodynamics are especially evident in Figure 5.22. Energy is first increased near the center of attraction by aligning the thrust vector near the velocity vector. Eccentricity is then increased by thrusting nearly perpendicular to the radius vector at large values of the radius.

Figure 5.24 shows several optimal orbital trajectories to a final eccentricity of .90. All of these transfers are initiated by a large gain in energy attained by proceeding along the minimum effort path. The space vehicle then increases its eccentricity at points far from the center of attraction. These trajectories again represent excellent examples of the fundamental principles of astrodynamics.

Figures 5.25 - 5.28 illustrate optimal orbital transfers to final orbits with a semi-major axis of one. In all cases only a change in final angular momentum is required. However, in accordance with the second fundamental principle, such a change is most economically performed at large radii. Consequently each of these transfers is initiated with an effort to increase the energy of the orbit. From Figure 5.28 it can be seen that this is done by proceeding along the path of minimum effort. When large radii are achieved, the thrust vector is pointed nearly perpendicular to the radius vector in

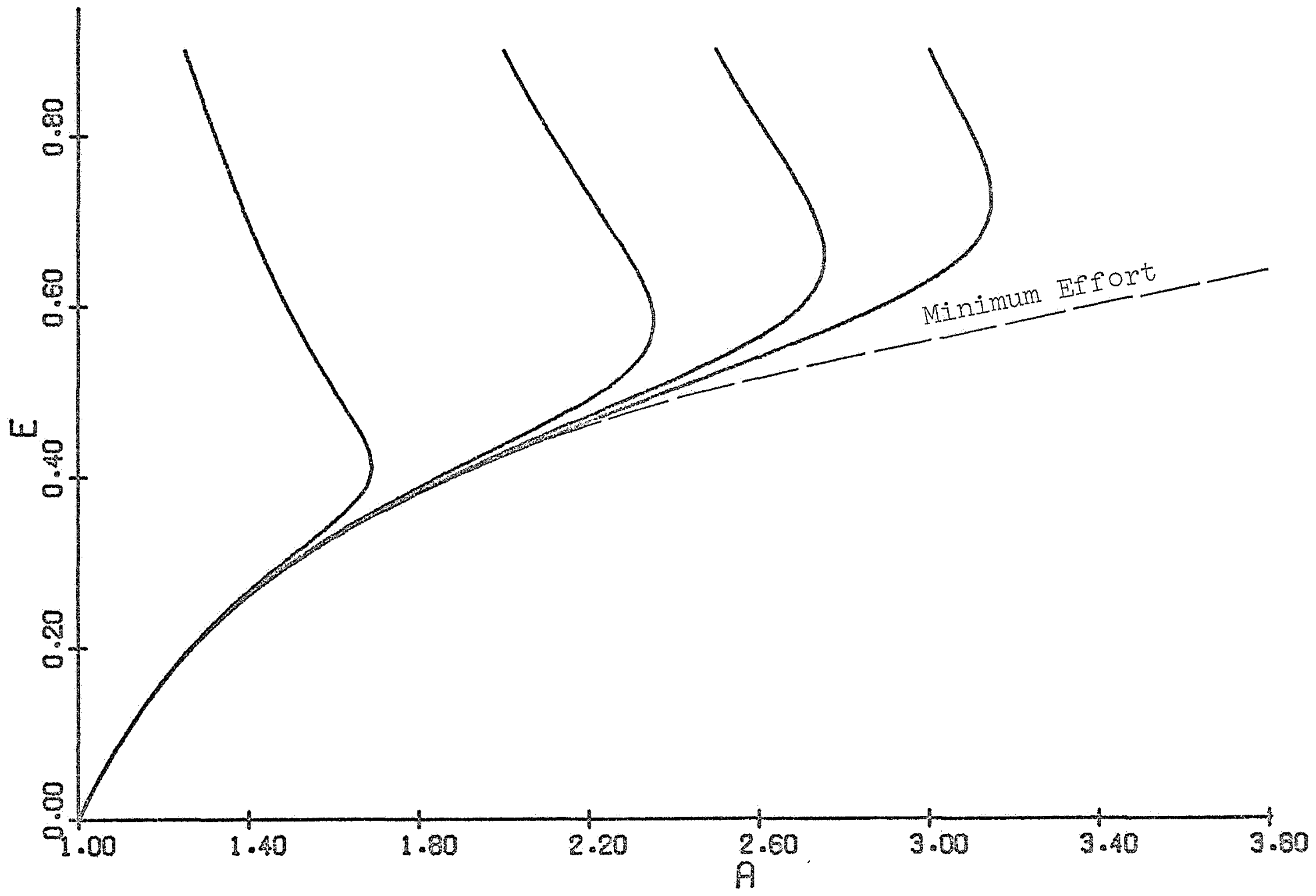

Fig. 5.24 Several Optimal Trajectories to Final Orbits with e = .90

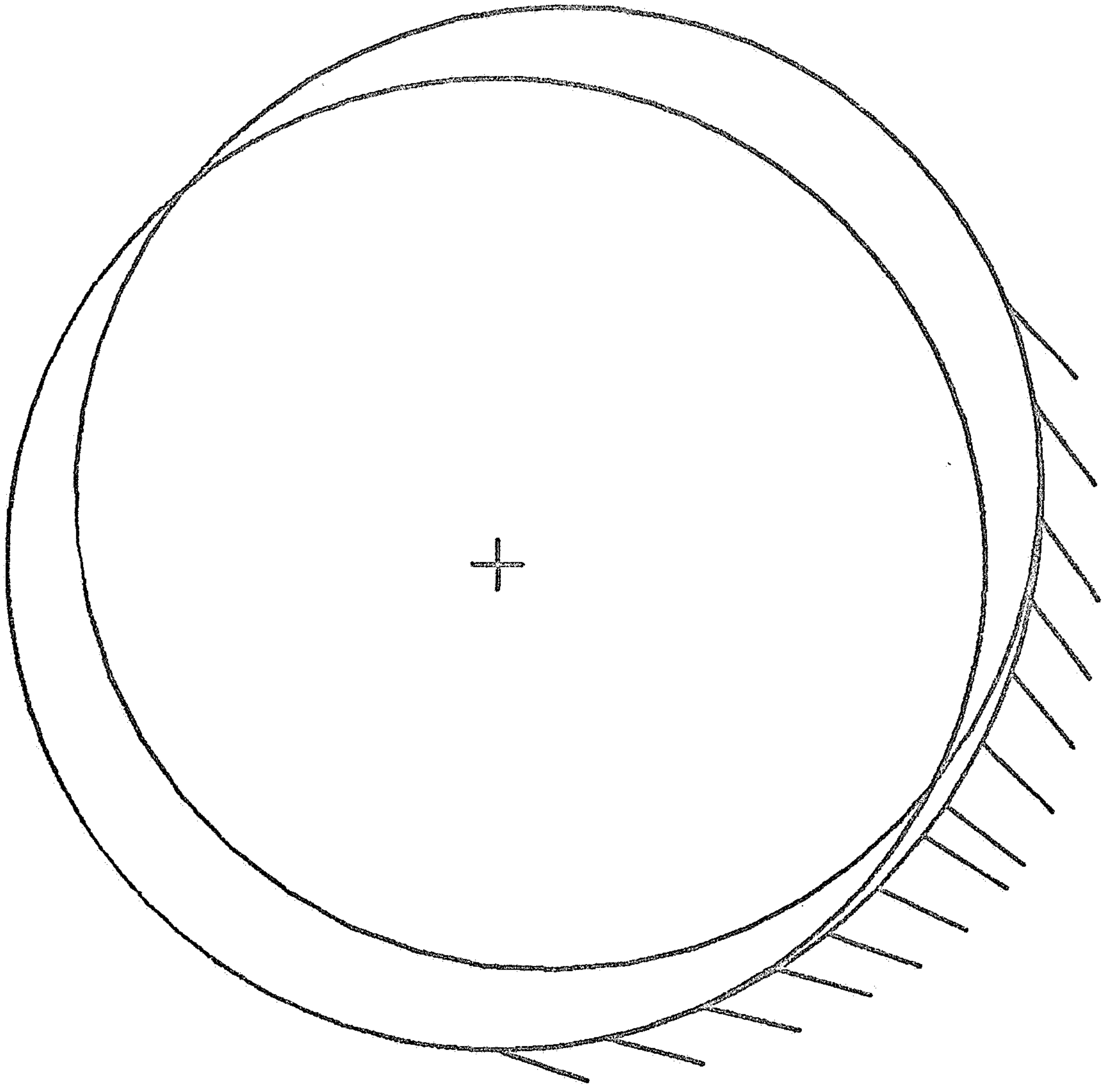

Fig. 5.25 An Optimal Orbital Transfer
to A = 1.0 and e = .20

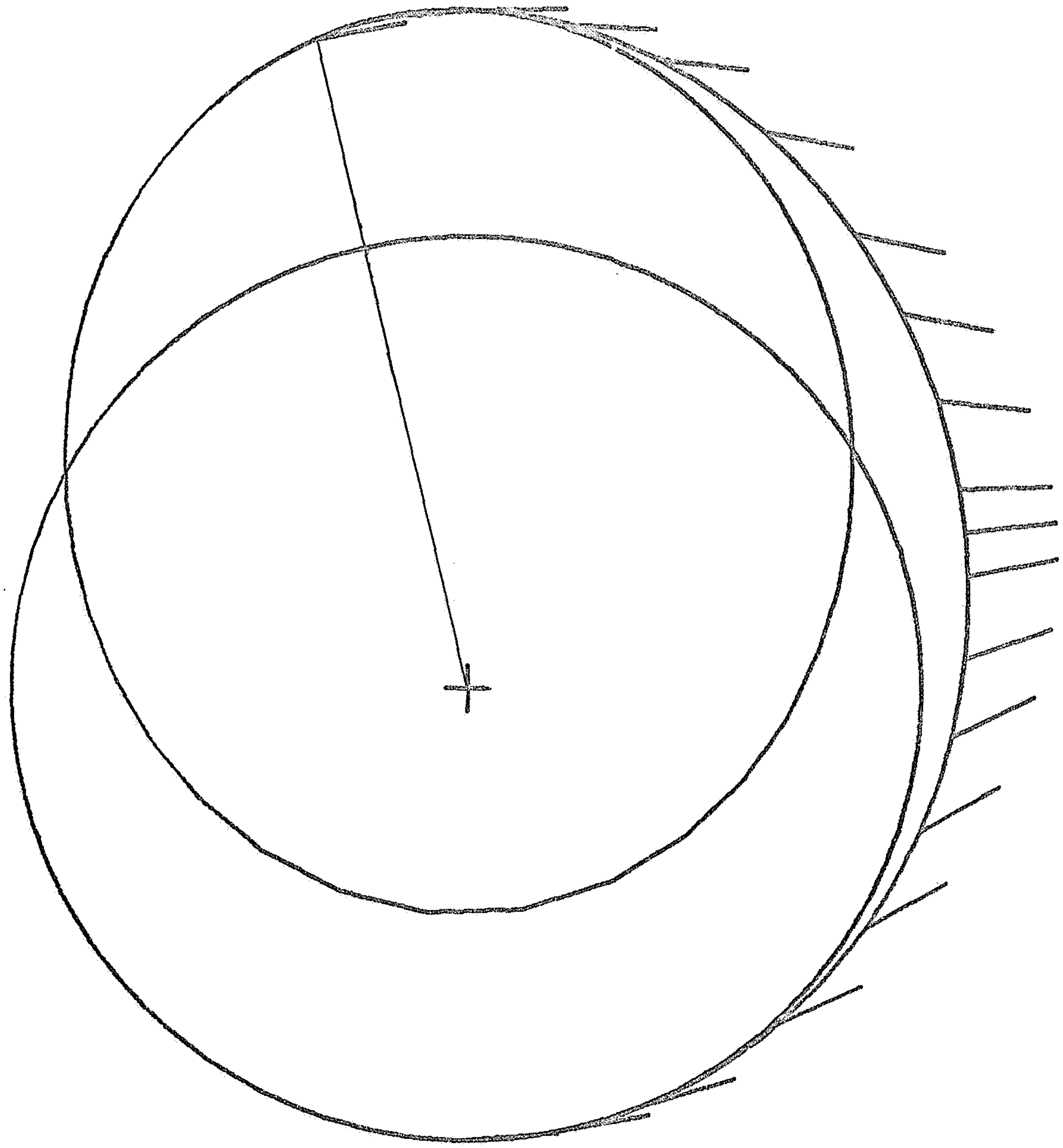

Fig. 5.26 An Optimal Orbital Transfer to A = 1.0 and e = .50

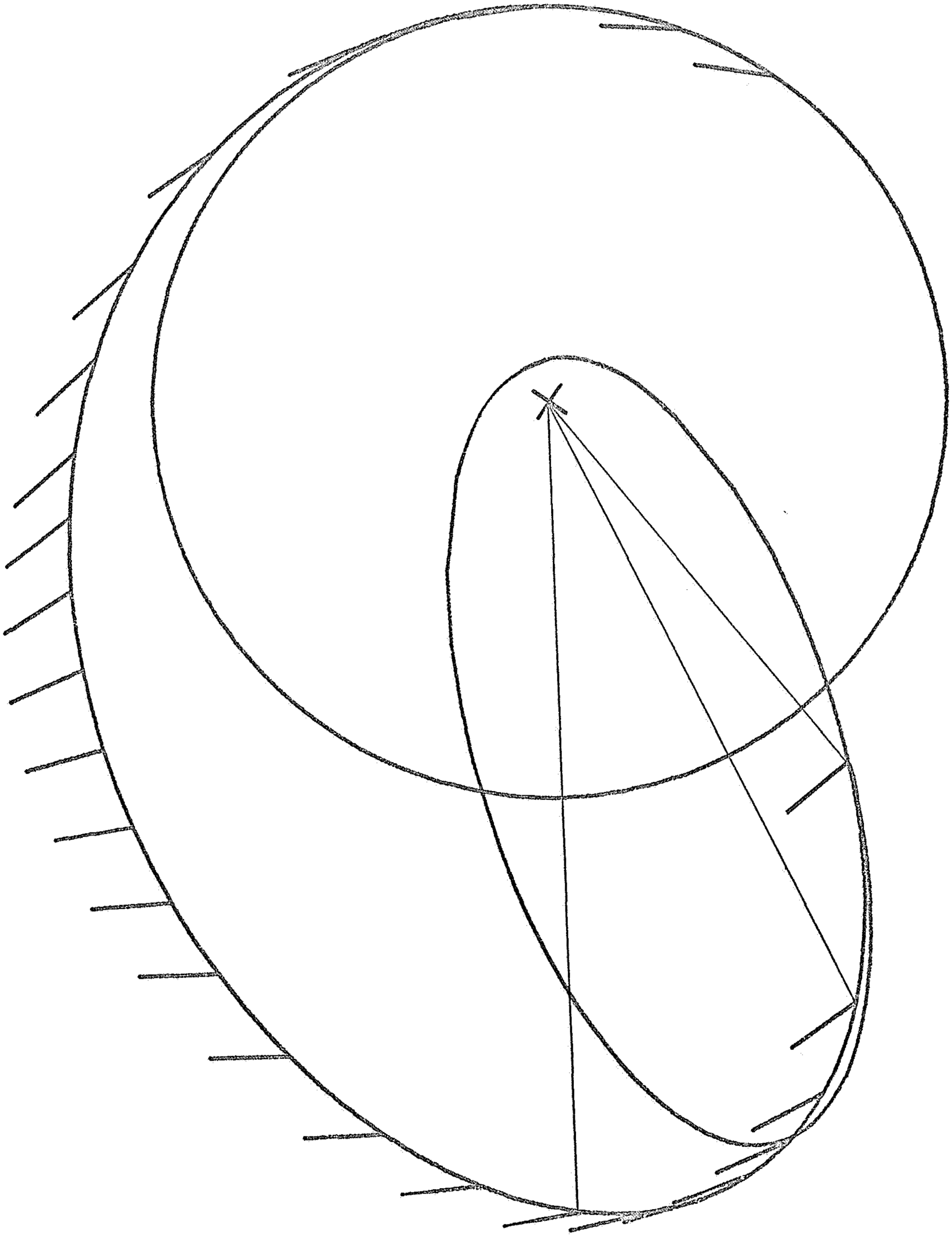

Fig. 5.27 An Optimal Orbital Transfer to A = 1.0 and e = .90

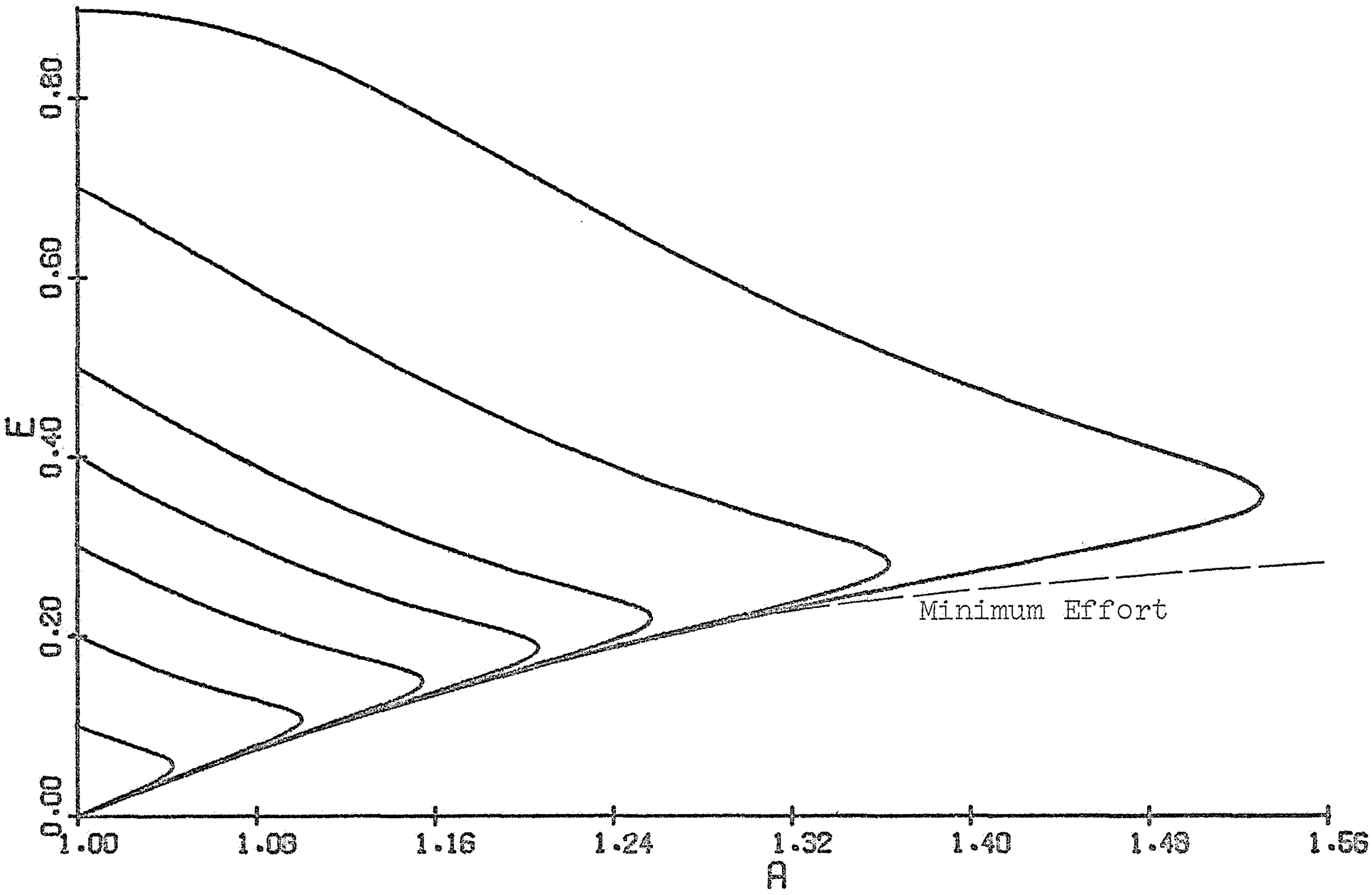

Fig. 5.28 Several Optimal Orbital Trajectories to A = 1.0

an effort to achieve the final angular momentum desired. At the same time a component of the thrust vector exists in the retro-fire position and the semi-major axis is decreased back to one.

Figures 5.29 - 5.32 illustrate orbital transfers to final orbits with semi-major axes of 0.75. A completely different region of transfers is encountered. To attain final orbits below the minimum effort curve (see Figure 5.32), the thrust vector is initially positioned in a retro-fire position and oscillates about $x = 180^{o}$. Two representative optimal transfers are shown in Figures 5.29 and 5.30.

In transferring to inner orbits with high eccentricity, it is difficult to lose the angular momentum required to achieve highly eccentric orbits, since the final semi-major axis must be small. In other words, the fundamental principle for losing energy is at odds with the principle for decreasing angular momentum. To achieve orbits with high eccentricity, the change of angular momentum required is the dominant factor. Consequently, to attain final orbits with eccentricities above the minimum effort path, the thrust vector is initially positioned with a component of thrust in the forward direction. The vehicle energy is first increased and as large radii are encountered, the energy and

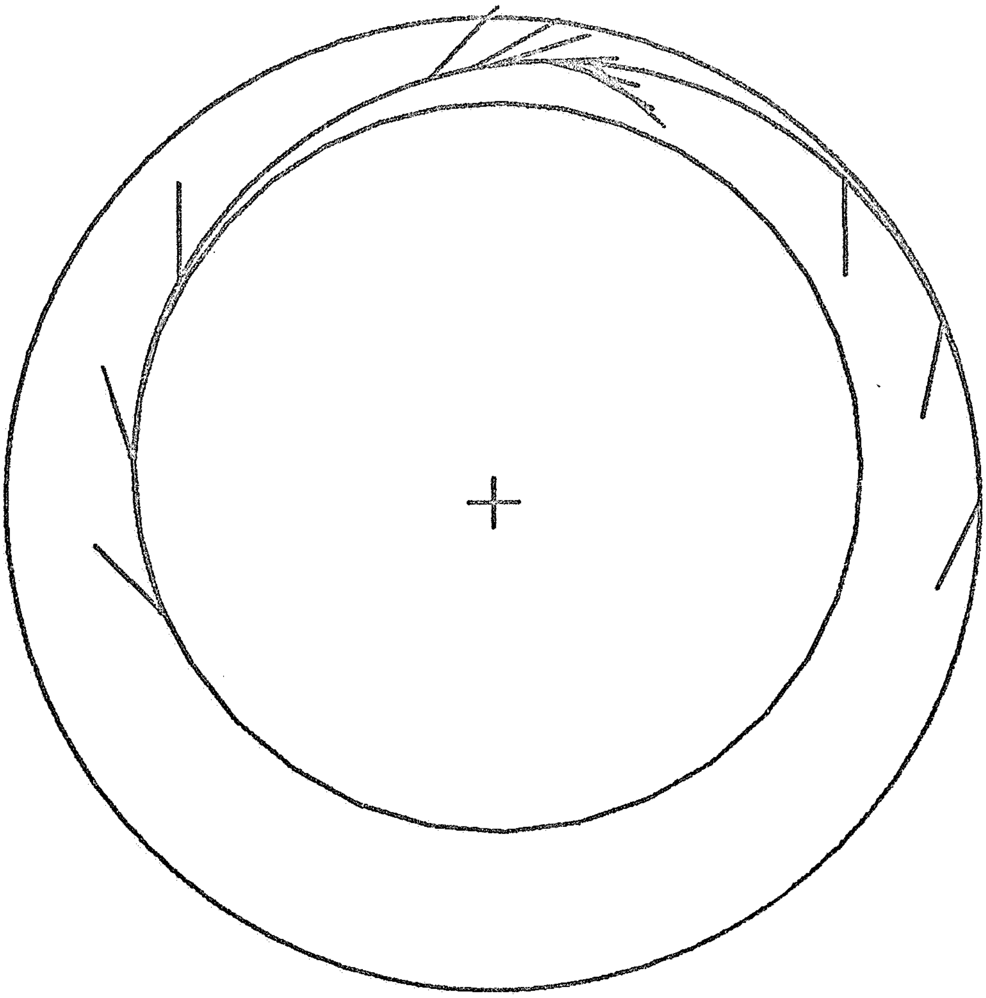

Fig. 5.29 An Optimal Orbital Transfer to A = .75 and e = .10

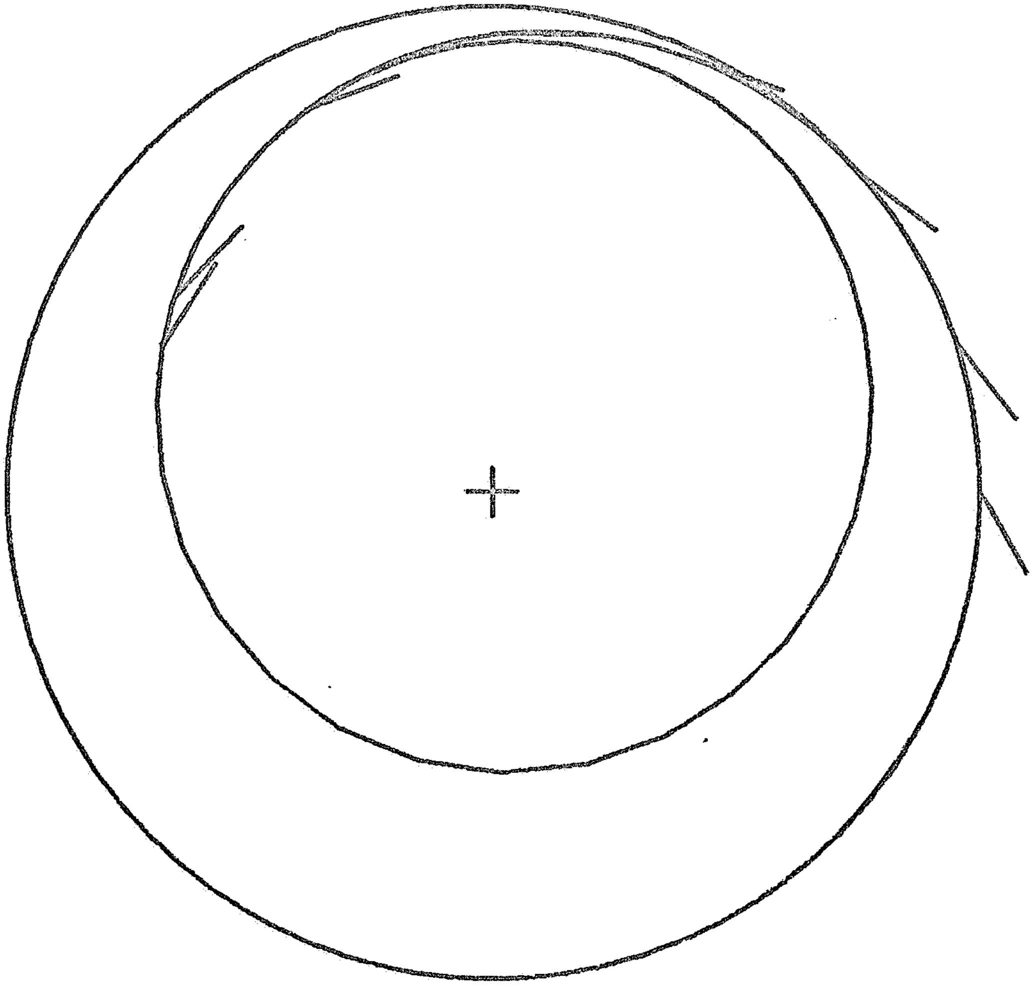

Fig. 5.30 An Optimal Orbital Transfer to A = .75 and e = .24

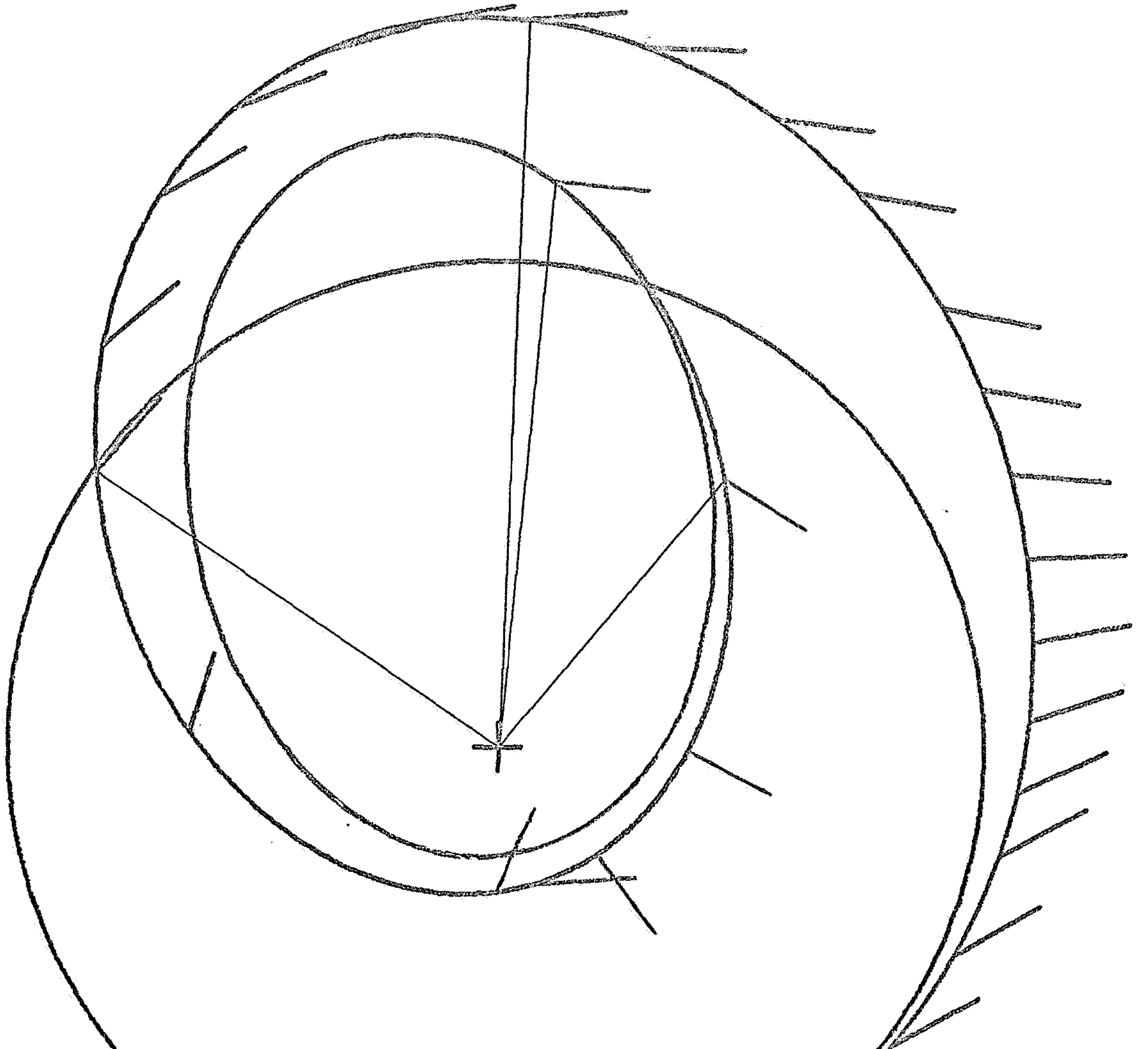

Fig. 5.31 An Optimal Orbital Transfer to A = .75 and e = .70

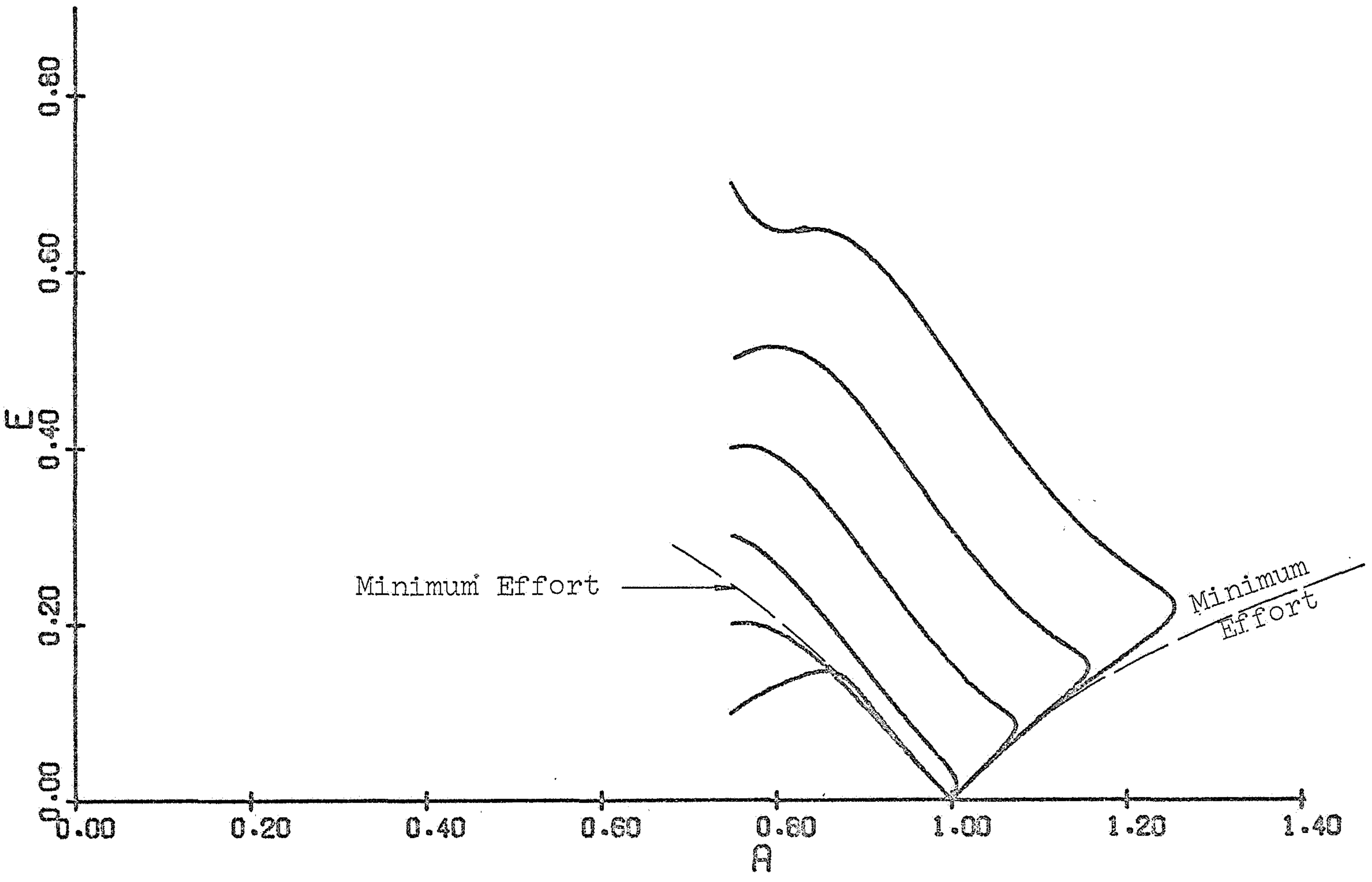

Fig. 5.32 Several Optimal Trajectories to A = .75

angular momentum are simultaneously decreased. Such a transfer is shown in Figure 5.31. In this case, thrust expended near the perigee of transfer is largely lost. If thrust magnitude were a control, the thrust would certainly have been turned off during this portion of transfer.

CHAPTER 6

CONCLUSIONS

Multiple stationary solutions are frequently obtained with optimization problems using the calculus of variations. The existence of multiple stationary solutions can be attributed to one of several distinct reasons. A reason that multiple solutions are frequently obtained is that the complete set of necessary and sufficient conditions for a solution have not been imposed.

It has been shown here that a sufficiency test for the Problem of Bolza can be broken down into two independent tests:

(1) a path sufficiency test, and

(2) endpoint sufficiency test.

An endpoint sufficiency test is developed here for problems with variable endpoints. If a solution candidate satisfies both test as well as the first three necessary conditions, it simply satisfies a sufficiency condition. That this sufficiency condition is also necessary has not been proven.

Multiple solutions can also occur in problems with singular control. To date the complete set of necessary and sufficient conditions needed to distinguish the true optimum have not been formulated.

Problems also exist for which the calculus of variations provides no criteria for selecting the global minimum. The theory of the calculus of variations is effective only in determining trajectories which are local optimums. Only small variation in the state space trajectory are considered. Problems with bounded control and periodic solution often have several local optima.

It has been shown that once the first necessary path conditions have been applied, a calculus of variations problem with variable endpoints is reduced to a problem of the minimization of a function of several variables. Analytical application of the endpoint sufficiency condition requires the analytical integration of the set of state variable and Euler-Lagrange differential equations. Since in most cases this is difficult or impossible, an algorithm has been developed for the numerical implementation of the endpoint sufficiency test.

The endpoint sufficiency condition has been shown to be an effective computational tool in complex applications. For example, through the use of the endpoint sufficiency test, a complete class of nonoptimal solutions can be discarded immediately upon encounter.

Without the aid of the endpoint sufficiency test an investigator would have no indication that solutions he is generating are nonoptimal until he encounters multiple solutions. This results in a great savings in computation time.

The alternative of determining the set of all multiple stationary solutions to a problem, so that the true optimal may be distinguished via a direct comparison of the performance indices, is often a formidable task. No criteria exist for determining the number of multiple solutions that exist and an investigator could never be sure he had found all of the multiple solutions.

The difficulty of solving a two-point boundary value problem needed to implement the endpoint sufficiency condition was eliminated by using the generalized Newton-Raphson algorithm. It has proved to be a powerful computational tool.

The comprehensive optimal orbital transfer example gave some interesting insights into the general problem of low thrust minimum fuel transfers. Qualitative aspects of all orbital transfers were found to be consistent with fundamental principles postulated for changing energy and angular momentum.

APPENDIX A

MINIMIZATION OF A FUNCTION OF SEVERAL VARIABLES

This appendix presents a rigorous proof of the necessary and sufficient conditions for a function of several variables to be a *minimum* when subject to algebraic equations of constraint. These proofs could be established easily, but with less rigor, through the use of Lagrange multipliers as was done in section 1.2. The validity of using such multipliers in establishing necessary conditions has been rigorously verified. However, the validity of using Lagrange multipliers to establish sufficiency conditions in control notation has not been established with rigor until recently (Vincent, 1969).

In the proof that follows, free use will be made of the notation and conventions established at the beginning of section 2.3. Following the methods of Vincent (1969), consider the problem of minimizing a function of several variables

$$J = J(\underline{w},\underline{v}) \tag{A.1}$$

subject to the constraints

$$\underline{\psi}\,(\underline{w},\underline{v}) = \underline{0} \tag{A.2}$$

where both J and $\underline{\psi}$ are functions of class C^2 and the constraints are such that the determinent of the Jacobian

$$\left[\frac{\partial\underline{\psi}}{\partial\underline{v}}\right] \qquad (A.3)$$

is nonsingular. The dimension of $\underline{\psi}$ and $\underline{w}$ is assumed to be p and the dimension of $\underline{v}$ is assumed to be q.

A.1 Method of Implicit Functions

Since $\underline{\psi}$ is of C^2 and condition (A.3) has been postulated, the implicit function theorem (Buck, 1965, pp. 283-286) states that equation (A.2) implicitly assures the existence of the vector function $\underline{W}$ explicitly relating the dependent variables $\underline{w}$ to the independent variables $\underline{v}$

$$\underline{w} = \underline{W}(\underline{v}) = \begin{vmatrix} W_1(\underline{v}) \\ W_2(\underline{v}) \\ \vdots \\ W_p(\underline{v}) \end{vmatrix} \qquad (A.4)$$

By substituting (A.4) into (A.1), J becomes a function of $\underline{v}$ only

$$J = J(W(\underline{v}),\underline{v}) \qquad (A.5)$$

Define the general value of independent variables $\underline{v}$ in a small neighborhood of an optional point $\underline{v}^o$

$$\underline{v} = v^{o} + \epsilon\ \underline{c} \tag{A.6}$$

where $\underline{c}$ is a vector of arbitrarily chosen small, but non-zero, constants and ϵ is a scaler multiplier. Then, from (A.1) and (A.2)

$$J = J\ [\underline{W}(\underline{v}^{o} + \epsilon\ \underline{c}),\ \underline{v}^{o} + \epsilon\ \underline{c}] \tag{A.7}$$

$$\psi\ [\underline{W}(\underline{v}^{o} + \epsilon\ \underline{c}),\ v^{o} + \epsilon\ \underline{c}] = 0 \tag{A.8}$$

Now J is a function of ϵ only, and the necessary condition for an ordinary local extremum is

$$\frac{dJ}{d\epsilon} = \frac{\partial J}{\partial \underline{w}}^{T} \underline{h} + \frac{\partial J}{\partial \underline{v}}^{T} \underline{c} = 0 \tag{A.9}$$

where $\underline{h}$ is the vector with elements $h_i = \frac{\partial W_i}{\partial \epsilon}$. The vector $\underline{h}$ represents changes in the dependent variables $\underline{w}$ corresponding to the changes $\underline{c}$ in the independent variables. Differentiating equation (A.8) with respect to ϵ yields

$$\left[\frac{\partial \underline{\psi}}{\partial \underline{w}}\right] \underline{h} + \left[\frac{\partial \underline{\psi}}{\partial \underline{v}}\right] \underline{c} = 0 \tag{A.10}$$

Solving for $\underline{h}$ yields

$$\underline{h} = - \left[\frac{\partial \underline{\psi}}{\partial \underline{w}}\right]^{-1} \left[\frac{\partial \underline{\psi}}{\partial \underline{v}}\right] \underline{c} \tag{A.11}$$

Substituting (A.11) into (A.9) and rearranging gives

$$\frac{dJ}{d\epsilon} = \left[\frac{\partial J}{\partial \underline{v}}^T - \frac{\partial J}{\partial \underline{w}}^T \left[\frac{\partial \underline{\psi}}{\partial \underline{w}} \right]^{-1} \left[\frac{\partial \underline{\psi}}{\partial \underline{v}} \right] \right] \underline{c} = \underline{0} \qquad (A.12)$$

Since $\underline{c}$ is an arbitrary nonzero vector, each of the elements of the vector in parenthesis must be equal to $\underline{0}$ at the optimal point.

$$\frac{\partial J}{\partial \underline{v}}^T - \frac{\partial J}{\partial \underline{w}}^T \left[\frac{\partial \underline{\psi}}{\partial \underline{w}} \right]^{-1} \left[\frac{\partial \underline{\psi}}{\partial \underline{v}} \right] = \underline{0} \qquad (A.13)$$

If equation (A.13) is satisfied, then a sufficient condition for a local minimum is that

$$\frac{d^2 J}{d\epsilon^2}_{v_o} \qquad (A.14)$$

must be positive definite for arbitrary values of $\underline{h}$ and $\underline{c}$ satisfying equation (A.11).

Before evaluating this expression, an identity for taking the partial derivative of the inverse of a matrix must be developed. Let A_{ij} represent a general element of the $\left[\frac{\partial \underline{\psi}}{\partial \underline{w}} \right]^{-1}$ matrix:

$$A_{ij} = \left[\frac{\partial \underline{\psi}}{\partial \underline{w}} \right]^{-1}_{ij} \qquad (A.15)$$

Then in indicial notation the definition of inverse may be expressed as

$$\delta_{qj} = \frac{\partial \psi_q}{\partial w_m} A_{mj} \qquad (A.16)$$

where δ_{qj} is the Kroniker delta. Premultiplying by A_{iq} gives

$$A_{ij} = A_{iq} \frac{\partial \psi_q}{\partial w_m} A_{mj} \qquad (A.17)$$

Taking the partial derivative of both sides yields

$$\frac{\partial}{\partial r_n} (A_{ij}) = \frac{\partial}{\partial r_n} (A_{iq}) \delta_{qj} + A_{iq} \frac{\partial}{\partial r_n} \left(\frac{\partial \psi_q}{\partial w_m} \right) A_{mj}$$

$$+ \delta_{im} \frac{\partial}{\partial r_n} (A_{mj}) \qquad (A.18)$$

Since δ_{ij} represents a constant, equation (A.18) reduces to

$$\frac{\partial}{\partial r_n} (A_{ij}) = \frac{\partial}{\partial r_n} (A_{ij}) + A_{iq} \frac{\partial}{\partial r_n} \left(\frac{\partial \psi_q}{\partial w_m} \right) A_{mj} + \frac{\partial}{\partial r_n} (A_{ij}) \qquad (A.19)$$

which gives the desired identity:

$$\frac{\partial}{\partial r_n} (A_{ij}) = - A_{iq} \frac{\partial}{\partial r_n} \left(\frac{\partial \psi_q}{\partial w_m} \right) A_{mj} \qquad (A.20)$$

From this point on results must be expressed in indicial notation since $\frac{\partial}{\partial r_n} \left(\frac{\partial \psi_q}{\partial w_m} \right)$ is a tensor. In indicial notation equation (A.12) becomes

$$\frac{dJ}{d\epsilon} = \left[\frac{\partial J}{\partial v_k} - \frac{\partial J}{\partial w_i} A_{ij} \frac{\partial \psi_j}{\partial v_k} \right] c_k \qquad (A.21)$$

Using equations (A.20) and (A.21) the sufficiency condition (A.14) yields

$$\frac{d^2J}{d\epsilon^2} = \frac{\partial^2 J}{\partial w_i \partial v_k} c_k h_i + \frac{\partial^2 J}{\partial v_n \partial v_k} c_k c_n$$

$$- \frac{\partial^2 J}{\partial w_n \partial w_i} A_{ij} \frac{\partial \psi_j}{\partial v_k} c_k h_n - \frac{\partial^2 J}{\partial v_n \partial w_i} A_{ij} \frac{\partial \psi_j}{\partial v_k} c_k c_n$$

$$+ \frac{\partial J}{\partial w_i} A_{iq} \frac{\partial^2 \psi_q}{\partial w_n \partial w_m} A_{mj} \frac{\partial \psi_j}{\partial v_k} c_k h_n \tag{A.22}$$

$$+ \frac{\partial J}{\partial w_i} A_{iq} \frac{\partial^2 \psi_q}{\partial v_n \partial w_m} A_{mj} \frac{\partial \psi_j}{\partial v_k} c_k c_n$$

$$- \frac{\partial J}{\partial w_i} A_{ij} \frac{\partial^2 \psi_j}{\partial w_p \partial v_k} c_k h_p$$

$$- \frac{\partial J}{\partial w_i} A_{ij} \frac{\partial^2 \psi_j}{\partial v_n \partial v_k} c_k c_n$$

must be positive definite for arbitrary values of $\underline{c}$ and $\underline{h}$ satisfying equation (A.11). Recognizing the indicial notation representation of equation (A.11)

$$h_m = - A_{mj} \frac{\partial \psi_j}{\partial v_k} c_k \tag{A.23}$$

in four terms of equation (A.22) and regrouping terms, condition (A.23) can be written

$$\frac{d^2J}{d\epsilon^2} = \left(\frac{\partial^2 J}{\partial v_n \partial v_k} - \frac{\partial J}{\partial w_i} A_{ij} \frac{\partial^2 \psi_j}{\partial v_n \partial v_k}\right) c_k c_n$$

$$+ \left(\frac{\partial^2 J}{\partial w_i \partial v_k} - \frac{\partial J}{\partial w_p} A_{pj} \frac{\partial^2 \psi_j}{\partial w_i \partial v_k}\right) c_k h_i$$

$$+ \left(\frac{\partial^2 J}{\partial v_n \partial w_i} - \frac{\partial J}{\partial w_p} A_{pq} \frac{\partial^2 \psi_q}{\partial v_n \partial w_i}\right) h_i c_n \qquad (A.24)$$

$$+ \left(\frac{\partial^2 J}{\partial w_j \partial w_i} - \frac{\partial J}{\partial w_p} A_{pq} \frac{\partial^2 \psi_q}{\partial w_i \partial w_j}\right) h_i h_j$$

must be positive definite for arbitrary values of c and h satisfying equation (A.11).

Equation (A.24) and equation (A.13) provide a set of necessary and sufficient conditions for $J(\underline{w},\underline{v})$ to be minimum when the equations of constraint $\underline{\psi}(\underline{w},\underline{v}) = 0$ must be satisfied.

A.2 Method of Lagrange Multipliers

The necessary and sufficient conditions can be put in a form which is more convenient to use by defining the augmented function

$$J^*(\underline{w},\underline{v},\underline{\mu}) = J(\underline{w},\underline{v}) + \underline{\mu}\ \underline{\psi}(\underline{w},\underline{v}) \qquad (A.25)$$

where $\underline{\mu}$ is a vector of constant multipliers called Lagrange multipliers.

If the Lagrange multipliers are given the identity

$$\mu_j = -\frac{\partial J}{\partial w_i} A_{ij} \qquad (A.26)$$

or in matrix notation,

$$\underline{\mu} = - \frac{\partial J}{\partial \underline{w}}^{T} \left[\frac{\partial \underline{\psi}}{\partial \underline{w}} \right]^{-1} \qquad (A.27)$$

several observations can be made. Thus necessary conditions for J to be a minimum (A.13) become

$$\frac{\partial J^*}{\partial \underline{v}} = \underline{0} \qquad (A.28)$$

The sufficiency conditions (A.24) may be written as

$$\left(\frac{\partial^2 J}{\partial v_n \partial v_k} + \mu_j \frac{\partial^2 \psi_j}{\partial v_n \partial v_k} \right) c_k c_n + \left(\frac{\partial^2 J}{\partial w_i \partial v_k} + \mu_j \frac{\partial^2 \psi_j}{\partial w_i \partial v_k} \right) c_k h_i$$

$$+ \left(\frac{\partial^2 J}{\partial v_n \partial w_k} + \mu_j \frac{\partial^2 \psi_j}{\partial v_n \partial w_i} \right) h_i c_n + \left(\frac{\partial^2 J}{\partial h_j \partial h_i} + \mu_q \frac{\partial^2 \psi_q}{\partial w_i \partial w_j} \right) h_i h_j \qquad (A.29)$$

must be positive definite

for arbitrary c and h satisfying equation (A.11). Define the vectors $\underline{r}$ and $\underline{d}$ as

$$\underline{r} = \left| \begin{array}{c} w_1 \\ w_2 \\ \vdots \\ w_p \\ v_1 \\ v_2 \\ \vdots \\ v_q \end{array} \right| \qquad \underline{d} = \left| \begin{array}{c} h_1 \\ h_2 \\ \vdots \\ h_p \\ c_1 \\ c_2 \\ \vdots \\ c_q \end{array} \right| \qquad (A.30)$$

The sufficiency condition (A.29) may now be put in compact matrix notation by using equation (A.25) and definitions (A.30):

$$\underline{d}^T \left[\frac{\partial^2 J*}{\partial \underline{r} \partial \underline{r}}\right] \underline{d} \qquad (A.31)$$

must be positive definite

for arbitrary $\underline{c}$ and $\underline{h}$ satisfying equation (A.11). To guarantee that $\underline{c}$ and $\underline{h}$ in vector $\underline{d}$ satisfy equation (A.11), elements of $\underline{d}$ may be expressed as functions of the independent constants $\underline{c}$ only. In matrix notation this may be expressed as

$$\underline{d} = \left[\begin{array}{c} \Phi \\ \hline I \end{array}\right] \underline{c} = \Omega \ \underline{c} \qquad (A.32)$$

where

$$\Phi = - \left[\frac{\partial \underline{\psi}}{\partial \underline{w}}\right]^{-1} \left[\frac{\partial \underline{\psi}}{\partial \underline{v}}\right] \qquad (A.33)$$

and I is a q by q identity matrix. The sufficiency condition (A.32) may then be written as

$$\underline{c}^T \Omega^T \left[\frac{\partial^2 J*}{\partial \underline{r} \partial \underline{r}}\right] \Omega \ \underline{c} \qquad (A.34)$$

must be positive definite

for arbitrary values of the elements of c.

The advantage of the Lagrange multiplier technique is that the necessary and sufficient comditions can be expressed in compact and easy-to-remember matrix notation.

APPENDIX B

PROGRAMS

```
C                         APPENDIX B
C
C
C     ***********************
C     * ORBITAL TRAJECTORIES *
C     ***********************
C
C     MAINLINE PROGRAM
C     PROGRAM SPACE CONTROLS THE GENERATION OF SOLUTIONS TO AN OPTIMAL
C     PROBLEM  IN CELESTIAL NAVIGATION. THE PROBLEM IS TO TRANSFER A LOW
C     THRUST SPACE VEHICLE FROM AN INITIAL CIRCULAR ORBIT TO ANY COPLANAR
C     ELLIPTIC ORBIT OF GIVEN ENERGY AND ANGULAR MOMENTUM.  THE THRUST
C     PROGRAM IS TO BE DETERMINED SO AS TO MINIMIZE THE  INTEGRAL OF THE
C     PRODUCT OF THRUST MAGNITUDE AND TIME.  THE FINAL ARG. OF PERIAPSIS
C     THE FINAL TRUE ANOMOLY, THE FINAL RANGE ANGLE, AND THE FINAL TIME
C     ALL ARE UNSPECIFIED AND CONSIDERED FREE.
C
C     THIS PROGRAM AND ASSOCIATED SUBROUTINES ARE EXPLAINED IN DETAIL IN
C     CHAPTERS 5 AND 3 AND IN SECTION 2.7 OF THIS DISSERTATION.
C     INPUT
C     *****
C     EACH SET OF INPUT DATA  REQUIRES TWO CARDS,
C     CARD 1
C     XI       = ESTIMATE OF INITIAL THRUST CONTROL ANGLE.
C     XDOTI    = ESTIMATE OF INITIAL NONDIMENSIONAL THRUST CONTROL
C                ANGLE RATE.
C     LRI      = ESTIMATE OF INITIAL RADIUS LAGRANGE MULTIPLIER.
C                NEEDED ONLY FOR PROBLEMS WITH MULTIPLE THRUSTING ARCS.
C     AF       = FINAL NONDIMENSIONAL SEMIMAJOR AXIS DESIRED.
C                NEEDED FOR FOR BOUNDARY VALUE PROBLEMS ONLY
C              = 0 IF THE FLOODING TECHNIQUE IS BEING USED AND NO
C                BOUNDARY VALUE PROBLEM IS TO BE SOLVED. (SECTION 3.1.1)
C     ECCF     = FINAL ECCENTRICITY DESIRED (FOR BOUNDARY VALUE PROBLEM)
C     TOL      = MAXIMUM INTEGRATION ERROR/ UNIT STEP. (APPROXIMATE ONLY)
C     CTO      = A CONVERGENCE TOLERANCE FOR TERMINATING ITERATION IN
C                SUBROUTINE QUASI DURING SOLUTION OF A BOUNDARY VALUE PR.
C     PARDEL   = PERCENTAGE CHANGE IN FINAL ENDPOINTS WHEN NUMERICALLY
C                COMPUTING PARTIAL DERIVATIVES FOR ENDPOINT SUFFICIENCY
C                TEST. (SEE SECTIONS 2.7 AND 3.3)
C
C     CARD2
C     NS       = RADIAL VELOCITY DIRECTION FOR ELLIPTICAL INITIAL ORBITS.
C              = -1 IF APROACHING PERIAPSIS
C     KFREQ    = MAXIMUM NUMBER OF INTEGRATION STEPS BETWEEN  PRINTED
C                TRAJECTORY OUTPUTS.
C     NV       = NUMBER OF THE VARIABLE WHICH IS TO BE CHANGED BEFORE
C                AUTOMATICALLY REPEATING A SOLUTION.
C              = 0 IF A NEW SET OF DATA CARDS IS TO BE READ IMMEDIATELY
C                  FOLLOWING THE SOLUTION GENERATED BY THIS CARD.
C              = 1 AND AF= 0, CHANGE XI     BY DV.
C              = 2 AND AF= 0, CHANGE XDOTI  BY DV.
C              = 3 AND AF= 0, CHANGE LRI    BY DV.
C              = 4 AND AF= 0, CHANGE FI     BY DV.
C              = 1 AND AF NOT EQUAL TO 0, CHANGE AF     BY DV.
C              = 2 AND AF NOT EQUAL TO 0, CHANGE ECCF   BY DV.
C              = 3 AND AF NOT EQUAL TO 0, CHANGE FI     BY DV.
C              = 4 AND AF NOT EQUAL TO 0, CHANGE UE     BY DV.
C     NCH      = NUMBER OF CHANGES, DV, TO BE MADE IN VARIABLE NV ABOVE.
C     NTYP     = 1 IF AN INITIAL CIRCULAR ORBIT AND A SINGLE THRUSTING
C                  PERIOD ARE DESIRED.
C              = 2 IF AN INITIAL CIRCULAR ORBIT AND MULTIPLE THRUSTING
C                  PERIODS ARE DESIRED. NOT FULLY DEBUGGED.
C              = 3 IF AN INITIAL ELLIPTIC ORBIT WITH MULTIPLE THRUSTING
```

```
C                   PERIODS IS DESIRED. THIS OPTION IS NOT FULLY DEBUGGED.
C     IF        = 1 IF AN ENDPOINT SUFFICIENCY CONDITION TEST IS TO BE
C                   PERFORMED.
C     NB        = NUMBER CF CUTOFFS.  THE SOLUTION IS TERMINATED WHEN THE
C                 NB-TH ZERO OF THE CUTOFF FUNCTION IS ENCOUNTERED.
C     ID        = DUMMY FOR FUTURE EXPANSION.
C     NIT       = MAXIMUM NUMBER OF GENERALIZED NEWTON-RAPHSON ITERATIONS
C                 ALLOWED FOR CONVERGENCE OF A SOLUTION TO THE BOUNDRY
C                 VALUE PROBLEM.
C     DV        = VALUE OF THE CHANGE TO BE MADE IN VARIABLE -NV- ABOVE.
C     PCT       = A MAXIMUM PERCENTAGE CHANGE ALLOWED IN INITIAL VALUES
C                 DURING THE SOLUTION OF A BOUNDARY VALUE PROBLEM.
C                 THIS PARAMETER IS USED TO CONTROL THE STABILITY OF THE
C                 CCNVERGENCE OF THE SOLUTION. (SEE SUBROUTINE QUASI)
C     FI        = MAXIMUM NONDIMENSICNAL THRUST MAGNITUDE.
C     UE        = NONDIMENSIONAL EXHAUST VELOCITY.
C
C
      PROGRAM SPACE(INPUT,OUTPUT,PUNCH,TAPE2=INPUT,TAPE3=OUTPUT,TAPE4=PU
     1NCH)
      REAL Y(100),P(20),CONST(10),XOLD(6,1000),YINIT(10)
      REAL M,LU,LG,LR,LM,LRI
      COMMON /MAIN/ E,H,SMA,ECC,CUT,S,APO,PER,X,WIER,DPR,HAM
      COMMON /INTVAR/ TIME,Y,P,NE,TOL,DX,MODE
      COMMON /IOLD/ XOLD
      EQUIVALENCE (Y(1),U), (Y(2),G), (Y(3),R), (Y(4),M), (Y(5),LU), (Y(
     16),LG), (Y(7),LR), (Y(8),LM), (Y(9),AN), (Y(10),TE)
      EXTERNAL DIFFI,BOUND,NONLIN
      EXTERNAL FIXED,CCNSTRT
      EXTERNAL SWITCH,CUTOFF
      DATA DPR/57.2957795131/
      DATA PI/3.141592653589793/
C
C          READ FIRST INPUT CARD. SEE COMMENTS ABOVE.
1     READ (2,2) XI,XDOTI,LRI,AF,ECCF,TOL,CTO,PARDEL
2     FORMAT (8E10.4)
C
C          READ SECOND INPUT CARD.  SEE COMMENTS ABOVE.
      READ (2,3) NS,NC,KFREQ,NV,NCH,NTYP,IF,NB,ID,ID,NIT,ID,DV,PCT,FI,UE
3     FORMAT (5I5,5I1,2I5,2F5.0,2F10.0)
      IF (FI.LE.0.0) STCP 1111
      NCS=NC
      NBS=NB
      ID=2
C
C          SET INITIAL DATA ASSUMING A CIRCULAR ORBIT WITH A SINGLE
C          THRUSTING PERIOD SPECIFIED.
4     U=1.0
      G=0.0
      R=1.0
      M=1.0
      LM=0.0
      E=-.5
      AN=0.0
      TE=0.0
      SMA=1.0
      X=XI
      XP=X
      XDOT=XDOTI
      LR=LRI
      F=FI
      X=X/DPR
      Y(11)=0.0
      Y(12)=0.0
      KSW=0
      UEST=1.0E+50
      FACTOR=-F*SIN(X)/M-U*XDOT+U*U*SIN(X)*SIN(G+X)/R+COS(G)/(R*R)
      IF (NTYP.GT.3) STOP
      GO TO (5,6,7), NTYP
C
C           CIRCULAR INITIAL ORBIT WITH NO MASS RATE AND A SINGLE THRUST.
C           X AND XDOT DETERMINE LU, LG, AND LR. LM IS NOT NEEDED.
5     LU=M*COS(X)
      LG=U*LU*SIN(X)/COS(X)
      LR=LG*FACTOR/(U*U*SIN(X)*COS(G+X))
      GO TO 8
C
C           CIRCULAR INITIAL ORBIT WITH FINITE MASS RATE.
C           X, XDOT AND LR DETERMINE LU, LG, AND LM.
6     LG=LR*U*U*SIN(X)*COS(G+X)/FACTOR
      LU=LG*COS(X)/(U*SIN(X))
      LM=(LU*(F*CCS(X)/M)+LG*(F*SIN(X)/(M*U)-1.0/(R*R*U)+U/R)-F)*UE/F
      GO TO 8
C
C           ELLIPTIC INITIAL ORBIT - START INTEGRATION AT PERIAPSIS.
C           H DETERMINES ORBIT SHAPE SINCE SMA IS NORMALIZED TO 1.
7     ECC=SQRT(1.0-H*H/SMA)
      R=SMA*(1.0-ECC)
      U=H/R
      GO TO 6
8     P(1)=F
      P(2)=UE
      S=SWITCH(TIME,Y,P)
      IF (S+0.00001.LT.0.0) P(1)=0.0
      CUT=CUTOFF(TIME,Y,P)
      H=R*U*COS(G)
C
C           SAVE INITIAL VALUES FOR USE IN B.V. PROBLEM.
      DO 9 I=1,10
9     YINIT(I)=Y(I)
C
C           PRINT OUT PRCBLEM PARAMETERS.
      XD=X*DPR
      WRITE (3,10) XD,XDOT,LR,F,H,UE,TOL,NS,NC,NTYP,NB
10    FORMAT (1H1,40X,21H***NEW TRAJECTORY*** //15H X, XDOT, LR = F13.8,
     12F13.8,30H   THRUST, MOM, EXH VEL, TOL =4F9.4,2X,4I2)
      WRITE (3,11)
11    FORMAT (//52H    X/ANGLE   ENERGY/A    MOM/ECC    VEL/LU     G/LG    5
     10HRAD/LRMASS/LMSWCH/CUTH/WIERAPO/PER//)
      CALL OUTPUT
      IF (AF.NE.0) GO TO 12
      PUNCH 36, F,UE,CNO,X,XDOT,U,G,R,LU,LG,LR,LM
      CNO=NC
C
C           INITIALIZE BOOKEEPING PARAMETERS FOR INTEGRATION.
12    KERR=0
      OS=0.0
      OCUT=0.0
      KF=KFREQ
      NB=NBS
      NC=NCS
      TIME=0.0
      NE=12
      MODE=0
      XNB=NB
      NLOOP=10.*XNB/TOL
```

```
C
C          INTEGRATION LOOP WITH TESTS FOR END OF TRAJECTORY, THRUST
C          CONTROL, PHYSICALLY IMPOSSIBLE SOLUTIONS, AND OUTPUT CONTROL.
C     *********************** START INTEGRATION LOOP *****************
      DO 23 MQ=1,NLCOP
      CALL RUNKUT (TIME,Y,P,NONLIN,NE,TOL,DX,MODE)
C
C          CHECK FOR SIGN CHANGE IN CUTOFF.  IF A CHANGE HAS OCCURRED,
C          CONVERGE STATE VARIABLES TO EXACT CUT OFF VALUE.
      CUT=CUTOFF(TIME,Y,P)
      IF (OCUT.EQ.0.0.OR.CUT*OCUT.GT.0.0.OR.MQ.LE.3) GO TO 15
      CALL CONVERG (CUTOFF,OCUT,CUT,DX,1.0E-08,NONLIN)
      MODE=0
      IF (CUT.EQ.0.0) 13,16
C
C          DETERMINE IF TRAJECTORY IS TO BE ENDED AT THIS ENDPOINT.
13    NB=NB-1
      WRITE (3,11)
      CALL OUTPUT
      WRITE (3,24)
      IF (KSW.EQ.1.AND.ABS(U-UEST).LT.0.02) GO TO 14
      IF (KSW.EQ.1) NB=NB+1
      IF (KSW.EQ.0.AND.NB.LE.0) 14,15
14    KERR=1
      GO TO 25
C
C          CHECK FOR SIGN CHANGE IN SWITCH.  IF A CHANGE HAS OCCURRED,
C          CONVERGE STATE VARIABLES TO EXACT SWITCHING VALUE.
15    IF (NC.LT.0) GO TC 20
      S=SWITCH(TIME,Y,P)
      IF (OS.EQ.0.0.OR.S*OS.GT.0.0.OR.MQ.LE.3) GO TO 20
      SNEW=S
      CALL CONVERG (SWITCH,OS,S,DX,1.0E-08,NONLIN)
      MODE=0
      IF (S.EQ.0.0) 17,16
16    KERR=2
      GO TO 25
C
C          TURN THRUST ON OR OFF DEPENDING ON SWITCH GOING + OR -.
17    NC=NC-1
      IF (NC.LT.0) GO TC 14
      IF (SNEW.GT.0.0) 19,18
18    P(1)=0.0
      WRITE (3,11)
      CALL OUTPUT
      GO TO 20
19    P(1)=F
      WRITE (3,11)
      CALL OUTPUT
C
C          OTHER HALTS BECAUSE OF PHYSICALLY IMPOSSIBLE SOLUTION.
20    IF (M*E.LT.0.0) GO TO 21
      KERR=4
      GO TO 25
C
C          PRINT OUT AFTER EVERY *KF* INCREMENTS OR AFTER EVERY TEN
C          DEGREES CHANGE IN X, WHICHEVER COMES FIRST.
21    X=ATAN(LG/(U*LU))
      KF=KF-1
      IF (ABS(X-XP).LT..174533.AND.KF.GT.0) GO TO 22
      KF=KFREQ
      XP=X
      CALL OUTPUT
C
22    OCUT=CUT
      OS=S
23    CONTINUE
      KERR=10
C     ************************* END INTEGRATION LOOP *****************
```

```
C
C         FINAL SUMMARY OUTPUT AFTER EACH SOLUTION.
24    FORMAT (1H )
25    FTA=ACOS((SMA*(1.0-ECC**2)/R-1.0)/ECC)
      IF (G.LT.0.0) FTA=-FTA
      AP=(AN-FTA)*DPR
      FTA=FTA*DPR
      ESTERR=TOL*TIME
      WRITE (3,26) FTA,AP,TE,KERR,MQ,TIME,ESTERR
26    FORMAT (/16H TRUE ANOMALY = F7.2,2X,20HARG. OF PERIAPSIS = F10.2,5
     1X,15HTOTAL EFFORT = F15.12//16H EXIT STATUS =  I4,5X,15HNO. OF STE
     2PS = I4,10H   TIME = ,F15.10,14H   EST. ERR = ,F8.6)
C         IF THE INTEGRATION JUST PERFORMED WAS A SOLUTION REGENERATION
C         THEN  JUMP.
      IF (KSW.EQ.1) GO TO 38
C         IF A BOUNDARY VALUE PROBLEM IS TO BE SOLVED USING THE FINAL
C         TIME JUST COMPUTED, THEN JUMP.
      IF (AF.NE.0.0) GO TO 27
      DD=NBS
      PUNCH 36, ECCI,TAI,E,H,SMA,ECC,AN,FTA,AP,TIME,TE,DD
C
C         AUTOMATIC INCREMENTATION OF INITIAL VARIABLES FOR ANOTHER
C         INTEGRATION.
      NCH=NCH-1
      IF (NCH.LT.0) GO TO 1
      IF (NV.EQ.1) XI=XI+DV
      IF (NV.EQ.2) XDOTI=XDOTI+DV
      IF (NV.EQ.3) LRI=LRI+DV
      IF (NV.EQ.4) FI=FI+DV
      GO TO 4
C
C         WITH TIME ESTIMATE KNOWN, REGENERATE SOLUTION WITH FIXED
C         STEP SIZE AND STORE IN XOLD AS ESTIMATE OF SOLUTION TO
C         THE BOUNDARY VALUE PROBLEM.
27    MQ=MQ*2
      IF (KERR.NE.1) GO TO 1
      IF (MQ.GT.1000) MQ=1000
      ESTIM=TIME
      DELT=ESTIM/MQ
      DO 28 I=1,10
28    Y(I)=YINIT(I)
      XOLD(1,1)=YINIT(1)
      XOLD(2,1)=YINIT(2)
      XOLD(3,1)=YINIT(3)
      XOLD(4,1)=YINIT(5)
      XOLD(5,1)=YINIT(6)
      XOLD(6,1)=YINIT(7)
      TIME=0.0
      MODE=5
      DO 29 I=2,MQ
      CALL RUNKUT (TIME,Y,P,NONLIN,NE,0.0,DELT,MODE)
      XOLD(1,I)=U
      XOLD(2,I)=G
      XOLD(3,I)=R
      XOLD(4,I)=LU
      XOLD(5,I)=LG
29    XOLD(6,I)=LR
C
C         SET UP FOR QUASILINEARIZATION SOLUTION OF B. V. PROBLEM.
30    P(1)=MQ
      P(2)=6
      P(3)=3
      P(4)=0
      P(5)=1
      P(6)=1
C
      P(19)=UE
      P(20)=F
C         SET UP FOR INTEGRATION OF MASS AS AN UNCOUPLED EQUATION.
```

```
      P(10)=1
      P(11)=1.0
C
C          THE CONSTRAINTS ARE ENERGY, ANGULAR MOMENTUM, LAMBDA TRANS.,
C          AND HAMILTONIAN = 0 DUE TO FREE TIME.
      CONST(1)=-1.0/(2.0*AF)
      CONST(2)=SQRT(AF*(1.0-ECCF**2))
      CONST(3)=0.0
      CONST(4)=0.0
      CALL QUASI (P,NIT,PCT,CTO,ESTIM,CONST,DIFFI,BOUND)
      IF (ESTIM.LT.0) 31,32
31    GO TO 1
C
C          SAVE THE INITIAL VALUES FOUND BY QUASI FOR THE REGENERATION
C          OF THE SOLUTION USING THE NONLINEAR EQUATIONS.
32    MSAVE=MQ
      U=XOLD(1,1)
      G=XOLD(2,1)
      R=XOLD(3,1)
      M=1.0
      LU=XOLD(4,1)
      LG=XOLD(5,1)
      LR=XOLD(6,1)
      LM=0.0
      UEST=XOLD(1,MQ)
C
C          END POINT SUFFICIENCY CONDITION TEST.
      IF (IF.NE.1) GO TO 35
      P(1)=MQ
      NIB=NIT
      CALL FOCAL (P,NIB,PCT,CTO,ESTIM,CONST,DIFFI,FIXED,CONSTRT,ID,PARDE
     1L)
      IF (ID.EQ.2) WRITE (3,33)
      IF (ID.NE.2) WRITE (3,34) ID
33    FORMAT(//20X, *END POINT SUFFICIENCY CONDITION SATISFIED.*/)
34    FORMAT(//20X, *END POINT SUFFICIENCY CONDITION NOT SATISFIED*, I3/
     2    20X,45H********************************************* //)
C
C          REGENERATE THE SOLUTION USING THE INITIAL VALUES FOUND BY
C          QUASI AND THE NONLINEAR DIFFERENTIAL EQUATIONS. THE PURPOSE
C          IS TO VERIFY THE NEWTON-RAPHSON RESULTS AND GENERATE A
C          COMPLETE LISTING OF THE TRAJECTORY.
35    X=ATAN(LG/(LU*U))
      IF (LU.LT.0.0) X=X+PI
      P(1)=F
      P(2)=UE
      E=-.5
      AN=0.0
      TE=0.0
      NB=1
      NC=NCS
      XDOT=-F*SIN(X)/(M*U)-LR*U*SIN(X)*COS(G+X)/LG+U*SIN(X)*SIN(G+X)/R+C
     1OS(G)/(U*R**2)
      DD=0.0
      TAI=0.0
      H=R*U*COS(G)
      ECCI=SQRT(1.0-H*H)
      CNO=NC
C
C          PUNCH OUTPUT - INITIAL CONDITIONS
      IF (ID.NE.2) GO TO 37
      PUNCH 36, F,UE,CNO,X,XDOT,U,G,R,LU,LG,LR,LM
36    FORMAT (6E13.6)
C          GO TO REGENERATE THE SOLUTION.
37    KSW=1
      GO TO 8
```

```
C
C         PUNCH OUTPUT - FINAL CONDITIONS
38    IF (ID.NE.2) GO TO 39
      DD=NB
      PUNCH 36, ECCI,TAI,E,H,SMA,ECC,AN,FTA,AP,TIME,TE,DD
C
C         AUTOMATIC INCREMENTATION OF INITIAL VARIABLES TO SOLVE A
C         BOUNDARY VALUE PROBLEM NEAR THE ONE JUST GENERATED.
39    MQ=MSAVE
      NCH=NCH-1
      IF (NCH.LT.0) GO TO 1
      IF (NV.EQ.1) AF=AF+DV
      IF (NV.EQ.2) ECCF=ECCF+DV
      IF (NV.EQ.3) F=F+DV
      IF (NV.EQ.4) UE=UE+DV
      GO TO 30
      END
C     *********************************
C     * COMPUTE PARAMETERS AND OUTPUT *
C     *********************************
C
C
      SUBROUTINE OUTPUT
      COMMON /MAIN/ E,H,SMA,ECC,CUT,S,APO,PER,X,WIER,DPR,HAM
      COMMON /INTVAR/ TIME,Y,P,NE,TOL,DX,MODE
      REAL Y(100),P(20)
      REAL M,LU,LG,LR,LM
      EQUIVALENCE (Y(1),U), (Y(2),G), (Y(3),R), (Y(4),M), (Y(5),LU), (Y(
     16),LG), (Y(7),LR), (Y(8),LM), (Y(9),AN), (Y(10),TE)
      EQUIVALENCE (P(1),F), (P(2),UE)
      DATA PI/3.141592653589793/
      E=.5*U*U-1.0/R
      H=R*U*COS(G)
      SMA=1.0/(2.0/R-U*U)
      ECC=SQRT(1.0-H*H/SMA)
      PER=SMA*(1.0-ECC)
      APO=SMA*(1.0+ECC)
      X=ATAN(LG/(U*LU))
      IF (LU.LT.0.0) X=X+PI
      HAM=LU*(F*COS(X)/M-SIN(G)/(R*R))+LG*(F*SIN(X)/(M*U)-COS(G)*(1.0/(R
     1*R*U)-U/R))+LR*U*SIN(G)-F*(1.0+LM/UE)
      WIER=-F*LU/(M*COS(X))
      XD=X*DPR
      GD=G*DPR
      AND=AN*DPR
1     IF (XD.LT.180.) GO TO 2
      XD=XD-360.
      GO TO 1
2     IF (XD.GT.-180.) GO TO 3
      XD=XD+360.
      GO TO 2
3     IF (AND.LT.180.) GO TO 4
      AND=AND-360.
      GO TO 3
4     IF (AND.GT.-180.) GO TO 5
      AND=AND+360.
      GO TO 4
5     WRITE (3,6) XD,E,H,U,GD,R,M,S,HAM,APO,AND,SMA,ECC,LU,LG,LR,LM,CUT,
     1WIER,PER
6     FORMAT (1H ,8F10.5,F10.7,F10.5/1H ,8F10.5,F10.7,F10.5/)
      RETURN
      END
```

```
C     ****************************************
C     * NORMAL-TANGENTIAL EQUATIONS OF MOTION*
C     ****************************************
C       Z = TIME, DX = STEP LENGTH, P = VECTOR OF PARAMETERS,
C                 Y  = DEP. VAR., DER = VECTOR OF DERIVATIVES OF Y-S.
      SUBROUTINE NONLIN (Z,DX,P)
      REAL P(20),Y(100),DER(100)
      REAL M,LU,LG,LR,LM
      COMMON /KUTTA/ DER,Y
      EQUIVALENCE (Y(1),U), (Y(2),G), (Y(3),R), (Y(4),M), (Y(5),LU), (Y(
     16),LG), (Y(7),LR), (Y(8),LM), (Y(9),AN), (Y(10),TE)
C
C     COMPUTE USEFUL COMBINATIONS.
      F=P(1)
      UE=P(2)
      OPP=LG/U
      HYPOT=1.0/SQRT(OPP**2+LU**2)
      XCOS=LU*HYPOT
      XSIN=OPP*HYPOT
      GCOS=COS(G)
      GSIN=SIN(G)
      FT=F*XCOS/M
      FN=F*XSIN/(M*U)
      RR=1.0/R
      DTA=U*GCOS*RR
      RR2=RR*RR
      RUR2=RR2/U
C
C     COMPUTE VELOCITY, FLIGHT PATH ANGLE, RADIUS AND MASS DERIVATIVES.
      DER(1)=FT-RR2*GSIN
      DER(2)=FN-RUR2*GCOS+DTA
      DER(3)=U*GSIN
      DER(4)=-F/UE
C
C     COMPUTE DERIVATIVES OF LAMBDA-S ASSOCIATED WITH ABOVE VARIABLES.
      DER(5)=LG*(FN-RUR2*GCOS-DTA)/U-LR*GSIN
      DER(6)=LU*RR2*GCOS-LG*(RUR2-U*RR)*GSIN-LR*U*GCOS
      DER(7)=-2.0*LU*RR2*RR*GSIN-LG*(2.0*RUR2*GCOS-DTA)*RR
      DER(8)=(LU*FT+LG*FN)/M
C
C     COMPUTE DERIVATIVES OF TRUE ANOMOLY, TOTAL EFFORT AND THRUST ANGLE
      DER(9)=DTA
      DER(10)=F
      DER(11)=0.0
      DER(12)=0.0
      RETURN
      END
C     ***************************************
C     DRIVE A FUNCTION OF AN INTEGRAL TO ZERO
C     ***************************************
C
C     CONFUN = FUNCTION SUBR. WHICH COMPUTES THE VALUE OF THE FUNCTION
C              TO BE DRIVEN TO ZERO.
C     OLD    = OLD VALUE OF THE FUNCTION BEFORE SIGN CHANGE.
C     NEW    = NEW VALUE OF THE FUNCTION AFTER SIGN CHANGE.
C     DXO    = INTEGRATION STEP LENGTH THAT CAUSED FUNCTION VALUE TO
C              CHANGE FROM OLD TO NEW.
C     ZTOL   = DESIRED MAXIMUM DEVIATION OF FUNCTION FROM ZERO.
C     DIFEQ  = SUBROUTINE WHICH COMPUTES DERIVATIVES OF STATES DESCRIB-
C              ING THE SYSTEM. SEE INTEGRATION SUBROUTINE *RUNKUT*.
C
      SUBROUTINE CONVERG (CONFUN,OLD,NEW,DXO,ZTOL,DIFEQ)
C     ******************
      COMMON /INTVAR/ TIME,Y,P,NE,TOL,DUMMY,DUM
      REAL Y(100),P(20),NEW
      MODE=5
      DX=DXO
      DO 1 I=1,10
      DX=NEW*DX/(CLD-NEW)
      OLD=NEW
      CALL RUNKUT (TIME,Y,P,DIFEQ,NE,TOL,DX,MODE)
      NEW=CONFUN(TIME,Y,P)
      IF (ABS(NEW).LT.ZTOL) 2,1
1     CONTINUE
      RETURN
2     OLD=0.0
      NEW=0.0
      RETURN
      END
      FUNCTION SWITCH (TIME,Y,P)
      REAL Y(100),P(20)
      SWITCH=SQRT((Y(6)/Y(1))**2+Y(5)**2)-Y(4)*(1.0+Y(8)/P(2))
      RETURN
      END
      FUNCTION CUTOFF (TIME,Y,P)
      REAL Y(100),P(20)
      CUTOFF=SQRT((Y(6)/Y(1))**2+Y(5)**2)-Y(4)
      RETURN
      END
C     ***********************************
C     QUASILINEARIZATION CONTROL SUBROUTINE *
C     ***********************************
C
C     BY R. G. BRUSCH     11/25/68
C
C     THIS SUBROUTINE CONTROLS THE GENERATION OF A SOLUTION TO A SET OF
C     N NONLINEAR ORDINARY FIRST ORDER DIFFERENTIAL EQUATIONS WITH
C     BOUNDARY CONDITIONS DESCRIBED AT THE FINAL AS WELL AS THE INITIAL
C     POINT.  THE SOLUTION TO THE NONLINEAR PROBLEM IS GENERATED AS THE
C     LIMIT OF A SERIES OF SOLUTIONS TO A CONNECTED LINEAR PROBLEM.
C
C     THE METHOD USED IS QUASILINEARIZATION.
C
C     REFERENCE, *A MODIFIED QUASILINEARIZATION METHOD FOR SOLVING
C                 TRAJECTORY OPTIMIZATION PROBLEMS*, JAY M. LEWALLEN,
C                 AIAA JOURNAL, VOL. 5, NO. 5, (MAY, 1967), PP 962-965.
C
C                *QUASILINEARIZATION AND NONLINEAR BOUNDARY VALUE
C                 PROBLEMS*, R. BELLMAN AND R. KALABA, 1965.
C
C                *SOLUTION OF VARIATIONAL PROBLEMS WITH BOUNDED CONTROL
C                 VARIABLES BY MEANS OF THE GENERALIZED NEWTON-RAPHSON
C                 METHOD*, BY P. KENNETH AND G. E. TAYLOR, IN *RECENT
C                 ADVANCES IN OPTIMIZATION TECHNIQUES*, EDITED BY LAVI
C                 AND VOGL.
C
C     DISCRIPTION OF VARIABLES
C     ***********************
C
C     PAR    = A VECTOR OF PARAMETERS USED IN INTEGRATION. 1-6 RESERVED.
C              THE FIRST 6 VALUES MUST CONTAIN THE FOLLOWING,     INDEX =
C
C     NS     = NO. OF STEPS IN TIME CURRENTLY BEING USED.               1
C     N      = NUMBER OF FIRST ORDER D.E. (1ST M HAVE KNOWN I.C.)       2
C            = NUMBER OF STATE VARIABLES
C     M      = NUMBER OF KNOWN STATE VARIABLE INITIAL CONDITIONS.       3
C     IA     = NUMBER OF *BANG-BANG* CONTROLS TIMES 3                   4
C     ITF    = 0, IF FINAL TIME IS FIXED. = 1 IF FINAL TIME IS FREE.    5
C     LD     = MAX. NO. OF TIME STEPS FOR WHICH STATES CAN BE STORED.   6
C
C
C     NIT    = MAXIMUM NUMBER OF COMPLETE ITERATIONS ALLOWED FOR CONVERG.
C     PC     = THE MAX. PERCENT CHANGE IN INITIAL VALUES TO BE ALLOWED.
C              THE PERCENTAGE IS BASED ON THE MAX. ABS. VALUE OF THE
C              VARIABLE OVER ITS RANGE. HIGH PCT YIELDS RAPID CONVERG.
C              LOW  PCT. LESSENS PROBAB. OF UNCONTROLABLE DIVERGENCE.
```

```
C     TOL    = MAX VALUE OF THE SUM OF (MAX. DEVIATIONS OF THE STATES
C                FROM THE STATES ON THE PREVIOUS ITERATION)
C                USED TO TEST FOR CONVERGENCE COMPLETION.
C     ESTIM = ESTIMATE OF FINAL TIME (ASSUMMED FREE)
C     CONST = A VECTOR CF CONSTANTS WHICH END POINT CONSTRAINTS MUST =.
C     DIFFI =
C     DIFFL = USER WRITTEN SUBROUTINES DEFINING LINEARIZED EQUATIONS
C     BOUND =         AND CONSTRAINT PROPERTIES.
C     X      = A VECTOR OF CURRENT STATE AND ALGEBRAIC VARIABLES
C              (N STATE VARIABLES FOLLOWED BY IA ALGEBRAIC VARIABLES).
C              AS SOLUTION IS DEVELOPED, SUCESSIVE VALUES (WRT. TIME) ARE
C              STORED IN XOLD MATRIX.
C     XOLD   = A MATRIX CF STATE AND ALGEBRAIC VARIABLES EVALUATED AT NS
C              TIME INTERVALS WHICH PROVIDES THE SUBROUTINE WITH AN
C              INITIAL SOLUTION WITH WHICH TO WORK.
C
C     USE
C     ***
C     THE USER DESCRIBES HIS PARTICULAR PROBLEM BY SUPPLYING SUBR. DIFFI
C     AND DIFFL, BY MODIFYING SUBR. BOUND, AND BY GENERATING A
C     REASONABLE APPROXIMATION TO THE SOLUTION (XOLD) AND BY CALLING
C     QUASI WITH THE PERTINENT SET OF PARAMETERS.
C
C
      SUBROUTINE QUASI (PAR,NIT,PC,TOL,ESTIM,CONST,DIFFI,BOUND)
C     ****************
      REAL X(100),CONST(10),UPDAT(10),DER(100),VARMAX(10)
      REAL XOLD(6,1000),PAR(20),ERR(10),DUMMY(100),NEWER
      COMMON /IOLD/ XOLD
      COMMON /KUTTA/ DER,DUMMY
      EXTERNAL DLSUB
C
C          SET UP CONSTANTS FOR THIS CALL.
      NS=PAR(1)
      N=PAR(2)
      M=PAR(3)
      IA=PAR(4)
      ITF=PAR(5)
      NA=PAR(10)
      NVECT=N-M
      NVAR=(NVECT+1)*N+NA
      NUP=N-M+ITF
      NST=N+IA
      DO 1 I=1,NUP
1     UPDAT(I)=0.0
      UPDAT(N-M+1)=0.0
      RHO=1.0E+50
      RHOLD=1.0E+50
      CMAX=1.0
      FRACT=0.0
      IDIV=0
C
C          TEST FOR AN IMPROPER CALL. IF YES, STOP.
      IF (N.LT.2) STOP 701
      IF (M.LT.1) STOP 702
      IF (NS.LT.1) STOP 703
      IF (M.GE.N) STOP 705
      IF (NIT.LT.1) STOP 706
      IF (PC.LE.0.0) STOP 707
      IF (TOL.LT.0.0) STOP 710
      IF (ESTIM.EQ.0.0) STOP 711
C
C          SET LOOP TO ITERATE NO MORE THAN *NIT* COMPLETE TIMES.
      KLZ=0
      DO 28 KRAP=KLZ,NIT
      IF (KRAP.EQ.0) GO TO 21
```

```
C
C           INITIALIZE AL AND BL MATRICES FOR CONSTANT ELEMENTS.
      CALL DIFFI (PAR)
C
C           SET UP INITIAL STATE VECTOR FOR GENERATING SIMULTANEOUSLY
C           A PARTICULAR SOLUTION TO THE INHOMCGENEOUS LINEARIZED
C           EQUATIONS (USING XOLD(I,1) VECTOR AS THE INITIAL CONDITION)
C           AND N-M SETS OF SOLUTION TO THE HOMOGENEOUS LINEARIZED
C           EQUATIONS (THE INITIAL CONDITIONS FOR EACH SET OF HOMO. EQ.
C           BEING ALL ZEROS EXCEPT A ONE IN PLACE OF THE INITIAL STATE
C           WHICH IS FREE TO VARY.)
C
      DO 2 I=1,NVAR
      IF (I.LE.N) X(I)=XOLD(I,1)
      IF (I.GT.N) X(I)=0.0
2     CONTINUE
      DO 3 I=1,NVECT
      J=I*N+M+I
3     X(J)=1.0
      IF (NA.EQ.0) GO TO 5
      DO 4 I=1,NA
4     X(NVAR-NA+I)=PAR(I+10)
C
C           INITIALIZE ERROR ESTIMATING VECTOR AND VECTOR USED TO SAVE
C           THE MAXIMUM VALUE OF THE VARIABLES REQUIRING INIT. GUESSES.
5     DO 6 I=1,N
6     ERR(I)=0.0
      DO 7 I=1,NVECT
7     VARMAX(I)=0.0
      TIME=0.0
      XNS=NS
      DELT=ESTIM/XNS
C
C           INTEGRATE ALL EQUATIONS FORWARD TO THE TIME ESTIMATE, ESTIM.
      DO 10 K=2,NS
      PAR(1)=K-1
C
C           INTEGRATE EQUATIONS ONE STEP.
C
C
      CALL RUNKUT (TIME,X,PAR,DLSUB,NVAR,0.0,DELT,5)
C
C     SAVE MAXIMUM ERROR IN EACH DEPENDENT VARIABLE FROM PARTICULAR
C     SOLUTION AND STORE NEW PARTICULAR SOLUTION FOR USE ON NEXT ITER.
      DO 8 I=1,N
      NEWER=ABS(X(I)-XOLD(I,K))
      IF (NEWER.GT.ERR(I)) ERR(I)=NEWER
8     XOLD(I,K)=X(I)
C
C           SAVE MAX. ABSOLUTE VALUE OF VARIABLES INITIALLY GUESSED
      DO 9 I=1,N
      IF (ABS(X(I)).GT.VARMAX(I)) VARMAX(I)=ABS(X(I))
9     CONTINUE
10    CONTINUE
C
C           COMPUTE ERROR METRIC
      RHO=0.0
      DO 11 I=1,N
11    RHO=RHO+ERR(I)
C
C           CHECK FOR HIGH DEVIATION OF INITIAL GUESS FROM EXACT NON-
C           LINEAR SOLUTION WITH SAME I.C.  IF TRUE, DO NOT CHANGE
C           INITIAL VALUES.  INTEGRATE ONCE  TOWARDS NONLINEAR SOLUTION.
      IF (KRAP.LE.1.AND.RHO.GT.10.*TOL) GO TO 21
C
C           CHECK FCR A DIVERGING SOLUTION.  ALLOW ONLY 15 CONSECUTIVE
```

```
C          DIVERGENCES.
      IF (RHOLD.EQ.0.0) GO TO 13
      ERATE=(RHOLD-RHO)/RHOLD
      IF (ERATE.GT.0.0) IDIV=0
      IF (ERATE.GT.0.0) GO TO 13
      IDIV=IDIV+1
      IF (IDIV.GE.15) WRITE (3,12)
   12 FORMAT(//// 10X, *QUASILINEARIZATION CONTROL SUBR. --- UNCONTROLAB
     2LE SYSTEM DIVERGENCE.*////)
      IF (IDIV.GE.15) GO TO 30
C
C          COMPUTE CHANGES IN INITIAL CONDITIONS WHICH WOULD SATISFY
C          B. C. IF THE SAME XOLD ESTIMATE WERE USED AGAIN.
   13 CALL BOUND (X,DER,CONST,PAR,UPDAT)
C
C          HOWEVER, ONLY USE A FRACTION OF THESE INITIAL VALUE UPDATES.
C          THIS WILL PRESERVE CONVERGENCE EVEN WITH BAD INIT. GUESSES.
C
C          IF SOLUTION IS CONVERGING MAKE ADJUSTMENTS TO INITIAL STATES
C          AND TIME ESTIMATE PROPORTIONAL TO FRACT.
C          FRACT IS CHOOSEN SO THAT NO CORRECTION WILL EXCEED  A PER
C          CENT OF THE MAX. ABSOLUTE VALUE OF THAT VARIABLE OVER RANGE.
      CMAX=1.0E-50
      DO 14 I=1,NVECT
      J=I+M
      IF (ABS(UPDAT(I)/VARMAX(J)).GT.CMAX) CMAX=ABS(UPDAT(I)/VARMAX(J))
   14 CONTINUE
      IF (ITF.EQ.1) 15,16
   15 CMAX=AMAX1(CMAX,ABS(UPDAT(NVECT+1)/ESTIM))
C
C          IF PERCENT ERROR IN LAST ITERATION .GT. PCT, REDUCE FRACT.
   16 DO 17 I=1,N
      IF (ABS(ERR(I)/VARMAX(I)).GT.CMAX) CMAX=ABS(ERR(I)/VARMAX(I))
   17 CONTINUE
      PCT=PC*0.01
      IF (CMAX.GT.PCT) FRACT=PCT/CMAX
      IF (CMAX.LT.PCT) FRACT=1.0
      IF (KRAP.GT.NIT-10.AND.FRACT.EQ.1.0) FRACT=0.8
C
C          UPDATE INITIAL VARIABLES.
      RHOLD=RHO
      IF (ITF.EQ.1) 18,19
   18 ESTIM=ESTIM+UPDAT(NUP)*FRACT
   19 DO 20 I=1,NVECT
      K=M+I
   20 XOLD(K,1)=XOLD(K,1)+UPDAT(I)*FRACT
C
C          TEST OUTPUT.
   21 ND=NS/9+1
      K=N+IA
      IF (KRAP.EQ.0) WRITE (3,22)
   22 FORMAT (1H1)
      WRITE (3,23) KRAP,RHO,FRACT,ESTIM,UPDAT(N-M+1)
   23 FORMAT(/*  ITERATION = *, I5, *  RHO = *, E12.3, *  FRACTION = *,
     2  F6.4, *  TIME EST. = *, E16.8,* TIME UPDATE = *,E12.4)
      DO 27 I=1,K
      IF (I.LE.M.OR.I.GT.N) 26,24
   24 WRITE (3,25) (XOLD(I,J),J=1,NS,ND),XOLD(I,NS),UPDAT(I-M)
   25 FORMAT (4X,1E16.8,8E10.2,1E16.8,1E10.2)
      GO TO 27
   26 WRITE (3,25) (XOLD(I,J),J=1,NS,ND),XOLD(I,NS)
   27 CONTINUE
C
C          CHECK FOR METRIC  WITHIN TOLERANCE. IF NOT REPEAT.
      IF (RHO.LT.TOL.AND.CMAX.LT.TOL.AND.KRAP.GT.2) GO TO 31
   28 CONTINUE
C
      WRITE (3,29)
   29 FORMAT(////10X, *QUASILINEARIZATION CONTROL SUBR. --- THE SYSTEM W
     2OULD NOT CONVERGE IN THE ALLOTTED NUMBER OF ITERATIONS.*////)
   30 ESTIM=-1.0
   31 PAR(1)=NS
      RETURN
      END
```

```
C     ****************************************
C     * FOURTH DEGREE RUNGE KUTTA INTEGRATION *
C     ****************************************
C
C     BY R. G. BRUSCH        2/13/68
C     MINIMUM TRUNCATION ERROR AS PER,
C     F. CESCHINO, NUMERICAL SOLUTION OF INITIAL VALUED PROBLEMS,
C     PAGES 44, 45, AND 67.
C
C     MODE = 5 FOR A FIXED STEP LENGTH =  DX
C          = 0 TO START INTEGRATION WITH STEP LENGTH SELECTED EVERY
C     4TH INTEGRATION SO ERR/UNIT TIME IS LESS THAN THE TOL.
C     NOTE THAT THIS DOES NOT NECESARILY GUARANTEE MINIMUM TOTAL ERROR
C     FOR THE NUMBER OF STEPS USED, SINCE ERROR PROPAGATION IS NOT
C     TAKEN INTO CONSIDERATICN.
C     MODE SHOULD NOT BE ADJUSTED BY THE MAIN LINE PROGRAM IN
C     THE VARIABLE STEP LENGTH MODE.
C
C
      SUBROUTINE RUNKUT (X,Y,P,DIF,NDE,TOL,DX,MODE)
      REAL X,Y(100),P(20),Y1(100),DER(100),XK(4,100),TOL,A(4,4),YOLD(3,1
     100),DEROLD(3,100),XS(4)
      COMMON /KUTTA/ DER,Y1
      DATA A(1,1),A(2,1),A(2,2)/0.40,-.14999999999999,0.750/,A(3,1),A(3,
     12),A(3,3)/.431818181818181,-.340909090909090,0.909090909090910/,A(
     24,1),A(4,2),A(4,3),A(4,4)/0.152777777777777,.347222222222221,.3472
     322222222221,.152777777777777/
C
C     CHECK MODE. IF INITIALLY ZERO, TAKE SMALL STEPS 4 TIMES TO SET
C     STEP LENGTH COMPATIBLE WITH SINGLE STEP ERROR TOLERANCE.
C                 IF INITIALLY 5, A FIXED STEP LENGTH =  DX IS USED.
      MQ=0
1     IF (MODE.GT.0) GO TO 2
      DX=.0001
      H=.0001
      HN=1.0E+50
      MODE=MODE-1
      GO TO 3
2     IF (MODE.EQ.5) H=DX
      IF (MODE.GT.5) STOP
C
C     SET THE INDEPENDENT VARIABLES
3     XS(1)=X
      XS(2)=X+H*0.4
      XS(3)=X+H*.6
      XS(4)=X+H
      L=IABS(MODE)
C
C     INTEGRATE - DIF IS A SUBROUTINE WHICH COMPUTES THE D/DX-S USING
C                 Y1-S, X, H, AND PARAMETERS P
      DO 4 J=1,NDE
4     Y1(J)=Y(J)
      DO 8 I=1,4
      CALL DIF (XS(I),H,P)
      DO 8 J=1,NDE
      XK(I,J)=DER(J)
      DER(J)=0.0
      DO 5 K=1,I
```

```
5     DER(J)=DER(J)+A(I,K)*XK(K,J)
      IF (I.LT.4) Y1(J)=Y(J)+H*DER(J)
      IF (I.LT.4) GO TO 8
      Y(J)=Y(J)+H*DER(J)
C
C     SAVE Y-S AND THEIR DERIVATIVES FOR ERROR EVALUATION AND H ADJUST.
      IF (L.EQ.5) GO TO 8
      IF (L.EQ.4) GO TO 6
      YOLD(L,J)=Y(J)
      DEROLD(L,J)=DER(J)
      GO TO 8
C
C     COMPUTE NEW H TO MAKE MAXIMUM ERROR LESS THAN THE TOLERANCE.
C     FOR 4TH ORDER RUNGE KUTTA, UNIT ERROR IS PROPORTIONAL TO H**4.
6     STEPER=+(.18333333333333*(Y(J)-YOLD(1,J))+.45*(YOLD(3,J)-YOLD(2,J)
     1))-(.05*(DER(J)+DEROLD(1,J))+.45*(DEROLD(3,J)+DEROLD(2,J)))*H
      IF (STEPER.EQ.0.0) GO TO 7
      UNITER=STEPER/H
      HNEW=H*(SQRT(ABS(TOL/UNITER)))
      IF (ABS(HNEW).LT.HN) HN=ABS(HNEW)
      IF (ABS(HNEW).LT.HN) MQ=J
      IF (ABS(HNEW).LT.HN) UN=UNITER
7     IF (J.LT.NDE) GO TO 8
      X=X+H-HN
      H=HN
      DX=H
      HN=1.0E+50
      MODE=0
8     CONTINUE
      X=X+H
      IF (MODE.LT.0) GO TO 1
      IF (MODE.LT.4) MODE=MODE+1
      RETURN
      END
C     ******************************************************************
C     COMPUTE DERIVATIVES FOR PARTICULAR SOLUTION AND N-M HOMO. SOLUTION
C     ******************************************************************
C
C
C     THE *N* LINEARIZED DIFFERENTIAL EQUATIONS ARE OF THE FORM,
C          D(X(J))/DT = AL(J,I)*X(I) + BL(I)                    (1)
C     WHERE REPEATED SUBSCRIPTS IMPLIES SUMMATION.
C     X(I)    = THE NEW ESTIMATE OF THE DEPENDENT VARIABLES.
C     AL(J,I) = A MATRIX OF COEFFICIENTS WHICH ARE EVALUATED USING ONLY
C               THE OLD STORED VALUES OF THE DEPENDENT VARIABLES.
C               THEY REPRESENT THE THE FIRST PARTIAL DERIVATIVES OF THE
C               RIGHT HAND SIDES OF THE NON-LINEAR D. E. WRT. THE STATE
C     BL(J)   = A VECTOR OF CONSTANT COEF WHICH ARE EVALUATED USING ONLY
C               THE OLD STORED VALUES OF THE DEPENDENT VARIABLES.
C
C
C     THE MATRIX   AL     AND VECTOR  BL     ARE FOUND FROM THE NON-LINE
C     EQUATIONS BY A TAYLOR SERIES EXPANSION OF THE RIGHT HAND SIDE OF T
C     1-ST ORDER NONLINEAR DIFFERENTIAL EQUATIONS ABOUT THE OLD STORED
C     SOLUTION AND RETAINS ONLY LINEAR TERMS OF THAT EXPANSION.
C     SINCE THE RESULTING EQUATIONS ARE LINEAR IN THE DEPENDENT VARIABLE
C     THE BOUNDARY CONDITIONS CAN BE SATISFIED IDENTICALLY AT EACH STEP
C     BY THE PRINCIPLE OF SUPERPOSITION.
C     THIS SUBROUTINE GETS THE VALUES FOR AL          AND BL    ONCE AND
C     USES THEM TO GENERATE THE DERIVATIVES OF THE INHOMOGENEOUS SOLUTIO
C     VECTOR AS WELL AS N-M HOMOGENEOUS SOLUTION VECTORS WHICH WILL BE
C     NEED FOR THE SUPERPOSITION.    THUS THE A AND B MATRICES ONLY
C     HAVE TO BE EVALUATED ONCE INSTEAD OF N-M+1 TIMES IF ALL OF THE
C     SOLUTIONS WERE NOT BEING GENERATED SIMULTANEOUSLY.
      SUBROUTINE DLSUB (T,DELT,PAR)
C     ****************
```

```
      REAL T,X(100),DER(100),PAR(20),AL(6,6),BL(6)
      COMMON /INTEG/ AL,BL
      COMMON /KUTTA/ DER,X
      IS=PAR(1)
      N=PAR(2)
      M=PAR(3)
      NVECT=N-M
C
C           GET CURRENT VALUE FOR AL, BL            MATRICES.
      CALL DIFFL (IS,PAR)
C
C           GET DERIVATIVES OF INHOMOGENEOUS EQUATIONS
      DO 2 I=1,N
      SUM=0.0
      DO 1 J=1,N
1     SUM=SUM+AL(I,J)*X(J)
2     DER(I)=SUM+BL(I)
C
C           GET DERIVATIVES OF *NVECT* SETS OF HOMOGENEOUS EQUATIONS.
      DO 5 K=1,NVECT
      KV=K*N
      DO 4 I=1,N
      SUM=0.0
      DO 3 J=1,N
3     SUM=SUM+AL(I,J)*X(J+KV)
4     DER(I+KV)=SUM
5     CONTINUE
C
C           USER INSERTS DERIVATIVES OF *PAR(10)* NO. OF AUXILLARY
C           VARIABLES WHICH ARE UNCOUPLED FROM THOSE INVOLVED IN THE
C           QUASILINEARIZATION. THE FIRST IS CALLED X((N-M+1)*N+1)
      DER(25)=-PAR(20)/PAR(19)
      RETURN
      END
C     **********************************
C     THE LINEARIZED DIFFERENTIAL EQUATIONS *
C     **********************************
C
C     USER WRITTEN ROUTINE TO EVALUATE AL AND BL MATRIX USING THE OLD
C     STORED VALUES.
C
C           PAR = A VECTOR OF CONSTANT PARAMETERS NEEDED FOR DERIVITIVE.
C           XOLD(I,J) = THE VALUE OF THE I-TH STATE AFTER J INTEGRATION
C                       STEPS ( FROM LAST ESTIMATE OF THE SOLUTION)
C
C
      SUBROUTINE DIFFL (IS,PAR)
      COMMON /INTEG/ AL,BL
      COMMON /IOLD/ XOLD
      COMMON /KUTTA/ DER,Y
      REAL AL(6,6),BL(6),PAR(20),XOLD(6,1000),DER(100),Y(100)
      REAL M,LU,LR,LG
      M=Y(25)
      UE=PAR(19)
      F=PAR(20)
      U=XOLD(1,IS)
      G=XOLD(2,IS)
      R=XOLD(3,IS)
      LU=XOLD(4,IS)
      LG=XOLD(5,IS)
      LR=XOLD(6,IS)
      COSG=COS(G)
C
C           COMPUTE FREQUENTLY APPEARING FACTORS.
      SING=SIN(G)
      FOM=F/M
      D=(LG*LG)/(U*U)+LU*LU
      SQRTD=SQRT(D)
      D32=SQRTD*D
C
C         UDOT EQUATION.
      AL(1,1)=FOM*LU*LG**2/(D32*U**3)
      AL(1,2)=-COSG/(R**2)
      AL(1,3)=2.0*SING/R**3
      AL(1,4)=FOM*(LG/U)**2/D32
      AL(1,5)=-FOM*LU*LG/(D32*U**2)
C
C         G DOT EQUATION.
      AL(2,1)=-FOM*LG*(LG**2+2.0*(LU*U)**2)/(D32*U**5)+(COSG/((R*U)**2)+
     1COSG/R)
      AL(2,2)=SING/(U*R**2)-U*SING/R
      AL(2,3)=(2.0/(U*R**3)-U/R**2)*COSG
      AL(2,4)=-FOM*LG*LU/(D32*U**2)
      AL(2,5)=FOM*(LU/U)**2/D32
C
C         RDOT EQUATION.
      AL(3,1)=SING
      AL(3,2)=U*COSG
C
C         LU DOT EQUATION.
      AL(4,1)=-FOM*(LG**2)*(2.0*(LG/U)**2+3.0*LU**2)/(D32*U**4)+2.0*LG*C
     1OSG/(U*(U*R)**2)
      AL(4,2)=LG*(1./(R*U)**2+1.0/R)*SING-LR*COSG
      AL(4,3)=LG*(2./(R*(R*U)**2)+1.0/R**2)*COSG
      AL(4,4)=-FOM*LG**2*LU/(D32*U**3)
      AL(4,5)=FOM*LG*((LG/U)**2+2.0*LU**2)/(D32*U**3)-(1.0/(R*U)**2+1.0/
     1R)*COSG
      AL(4,6)=-SING
C
C         LG DOT EQUATION.
      AL(5,1)=LG*(1.0/((R*U)**2)+1./R)*SING-LR*COSG
      AL(5,2)=(LR*U-LU/R**2)*SING-LG*(1.0/(R*R*U)-U/R)*COSG
      AL(5,3)=-2.0*LU*COSG/R**3+LG*(2.0/(U*R**3)-U/R**2)*SING
      AL(5,4)=COSG/R**2
      AL(5,5)=-(1.0/(U*R**2)-U/R)*SING
      AL(5,6)=-U*COSG
C
C         LR DOT EQUATION.
      AL(6,1)=LG*(2.0/(R*(R*U)**2)+1.0/R**2)*COSG
      AL(6,2)=AL(5,3)
      AL(6,3)=6.0*LU*SING/R**4+LG*(6.0/(U*R**4)-2.*U/R**3)*COSG
      AL(6,4)=-2.0*SING/R**3
      AL(6,5)=-(2.0/(U*R**3)-U/R**2)*COSG
C
C         NON-LINEAR TERMS.
      BL(1)=FOM*LU**3/D32-3.0*SING/R**2+G*COSG/R**2
      BL(2)=FOM*LG*(2.*(LG/U)**2+3.0*LU**2)/(D32*U**2)+(-4.0/(U*R**2)+U/
     1R)*COSG-G*AL(2,2)
      BL(3)=-U*G*COSG
      BL(4)=FOM*LG**2*(2.0*(LG/U)**2+3.0*LU**2)/(D32*U**3)+LG*(-4.0/(R*U
     1)**2-1.0/R)*COSG-G*AL(4,2)
      BL(5)=-3.0*LG*SING/(U*R**2)+(2.0*LU/R**2+LR*U)*COSG-G*AL(5,2)
      BL(6)=-LG*(8./(U*R**3)-U/R**2)*COSG-6.*LU*SING/R**3-G*AL(6,2)
      RETURN
      END
```

```
C     ***********************************************************
C     INITIALIZE AL AND BL     MATRICES FOR CONST. ELEMENTS
C     ***********************************************************
C
C
      SUBROUTINE DIFFI (PAR)
      COMMON /IOLD/ XOLD
      COMMON /INTEG/ AL,BL
      REAL AL(6,6),BL(6),PAR(20),XOLD(6,1000)
      N=PAR(2)
      M=PAR(3)
C
      DO 1 I=1,N
      BL(I)=0.0
      DO 1 J=1,N
1     AL(I,J)=0.0
C
      RETURN
      END
C     ****************************************
C     END  POINT SUFFICIENCY CONDITION TEST   *
C     ****************************************
C
C
      SUBROUTINE FOCAL (PAR,NIT,PCT,TOL,ESTIM,CONST,DIFFI,FIXED,CONSTRT,
     1ID,D)
      REAL XOLD(6,1000),PAR(20),CONST(10),SAV(10)
      REAL PARTWO(10,10),PHI(10,10),A(10,10),XS(10)
      INTEGER ICONST(10)
      COMMON /IOLD/ XOLD
      NS=PAR(1)
      N=PAR(2)
      M=PAR(3)
      ITF=PAR(5)
      NTC=N-M+ITF
C
C          GET NO. OF VARIABLES INVOLVED IN CONSTRAINTS, NO. OF
C          INDEPENDENT VARIABLES INVOLVED IN CONSTRAINTS, LOCATIONS OF
C          EACH, 2ND PARTIAL DERIVATIVES OF G, AND PHI CONSTRAINT MATRIX
      CALL CONSTRT (L,PAR,NV,NI,ICONST,PARTWO,PHI)
C
C          GENERATE PARTIAL DERIVATIVES OF LAMBDA FINALS WRT. DEPENDENT
C          AND INDEPENDENT VARIABLES INVOLVED IN CONSTRAINTS.
      WRITE (3,1) ((PARTWO(J,I),I=1,NV),J=1,NV)
1     FORMAT (3E16.8)
      WRITE (3,2) ((PHI(I,J),J=1,NI),I=1,NV)
2     FORMAT (/(E16.8))
      DO 3 I=1,N
3     SAV(I)=XOLD(I,NS)
      TIMSAV=ESTIM
      DO 4 I=1,N
4     XS(I)=XOLD(I,1)
      IF (NV.GT.NTC) STOP 40
      DO 10 I=1,NV
      DO 7 J=1,NV
      CONST(J)=SAV(J)
      IF (I.EQ.J) 5,7
5     CONST(J)=SAV(J)*(1.0+D)
      DENOM=SAV(J)*D
C
C          TEST FOR ENDPOINTS NEAR ZERO.
      IF (ABS(CONST(J)).LT..001) 6,7
6     CONST(J)=SAV(J)+.01
      DENOM=.01
7     CONTINUE
      CALL QUASI (PAR,NIT,PCT,TOL,ESTIM,CONST,DIFFI,FIXED)
      IF (ESTIM.LT.0.0) ID=0
      IF (ESTIM.LT.0.0) RETURN
      NH=N/2
      K=0
      DO 9 L=1,NH
      IF (ICONST(L).NE.0) 8,9
8     K=K+1
```

```
      PARTWO(K,I)=PARTWC(K,I)+(XOLD(L+NH,NS)-SAV(L+NH))/DENOM
9     CONTINUE
      IF (K.GT.NV) STOP 42
      ESTIM=TIMSAV
      DO 10 K=1,N
10    XOLD(K,1)=XS(K)
      WRITE (3,1) ((PARTWO(J,I),I=1,NV),J=1,NV)
C
C           POST- AND PRE-MULTIPLY BY PHI
      DO 12 I=1,NV
      DO 12 J=1,NI
      SUM=0.0
      DO 11 K=1,NV
11    SUM=SUM+PARTWO(I,K)*PHI(K,J)
12    A(I,J)=SUM
      DO 14 I=1,NI
      DO 14 J=1,NI
      SUM=0.0
      DO 13 K=1,NV
13    SUM=SUM+PHI(K,I)*A(K,J)
14    PARTWO(I,J)=SUM
      WRITE (3,2) ((PARTWO(J,I),I=1,NI),J=1,NI)
C
C           CHECK DEFINITENESS.
      CALL DEFINIT (PARTWO,NI,ID)
      RETURN
      END
C     ***************************************
C     COMPUTATICN OF REAL VARIABLE CONSTRAINTS  *
C     ***************************************
C     KS   = 1 FOR A NCRMAL CALL FROM FOCAL
C     PAR  = A VECTOR OF PARAMETERS AS DESCRIBED IN QUASI.
C     NVAR = THE NUMBER OF VARIABLES INVOLVED IN THE CONSTRAINTS.
C     NI   = THE NUMBER OF INDEPENDENT VARIABLES INVOLVED IN CONSTRAINTS
C     NC   = NO. OF CCNSTRAINED VARIABLES = NO. OF CONSTRAINTS.
C     ICONST      = 1 IF X(J) IS A DEPENDENT VARIABLE IN A CONSTRAINT.
C                 = 2 IF X(J) IS AN UNCONSTRAINED VARIABLE IN CONSTRT.
C                 = 0 OTHERWISE
C     PARTWO(I,J) = VALUE OF SECOND PARTIAL DERIVATIVE OF G WITH RESPECT
C                   TO THE I-TH AND J-TH VARIABLE INVOLVED IN THE
C                   CONSTRAINTS.  THE MATRIX IS ORDERED SO THAT ALL
C                   DERIVATIVES WRT THE DEPENDENT VARIABLES APPEAR IN
C                   THE UPPER-LEFT. (WITH I AND J .LT. NC)
C     PHI(I,J)    = A MATRIX RELATING CONSTRAINED VARIABLES TO THE
C                   UNCONSTRAINED VARIABLES. CON(I) = PHI(I,J)*UNCON(J)
C
C
      SUBROUTINE CONSTRT (KS,PAR,NVAR,NI,ICONST,PARTWO,PHI)
C     ******************
      REAL PAR(20),PARTWO(10,10),PHI(10,10),XOLD(6,1000),PARCD(10,10)
      REAL PARCI(10,10),PARCDI(10,10),X(10)
      INTEGER ICONST(10)
      COMMON /IOLD/ XOLD
      REAL MU1,MU2
      REAL LG,LR
      NS=PAR(1)
      N=PAR(2)
C     USER LOADS ICONST, PARTWO, PARCD, AND PARCI HERE.
C     ***************************************************
C
C     CHOOSE U AND G TO BE INDEPENDENT
      U=XOLD(1,NS)
      G=XOLD(2,NS)
      R=XOLD(3,NS)
      LG=XOLD(5,NS)
      LR=XOLD(6,NS)
      MU1=-R*R*(LG/(R*TAN(G))+LR)
      MU2=LG/(R*U*SIN(G))
      PARTWO(1,1)=MU1
      PARTWO(2,2)=-MU2*U*R*COS(G)
      PARTWO(3,3)=-2.0*MU1/R**3
      PARTWO(1,2)=PARTWO(2,1)=-MU2*R*SIN(G)
      PARTWO(1,3)=PARTWO(3,1)=MU2*COS(G)
      PARTWO(2,3)=PARTWO(3,2)=-MU2*U*SIN(G)
      ICONST(1)=1
      ICONST(2)=1
      ICONST(3)=2
      ICONST(4)=0
      ICONST(5)=0
      ICONST(6)=0
C
C          COMPUTE PARTIAL DERIVATIVES OF CONSTRAINTS WRT. DEPENDENT VAR
      PARCD(1,1)=U
      PARCD(1,2)=0.0
      PARCD(2,1)=R*COS(G)
      PARCD(2,2)=-R*U*SIN(G)
C
C     COMPUTE PARTIALS OF CONSTRAINTS WRT. INDEPENDENT VARIABLES.
      PARCI(1,1)=1.0/R**2
      PARCI(2,1)=U*COS(G)
C     *************************************************
C
C          SET NC, NVAR, AND NI.
      NC=0
      NVAR=0
      NI=0
      DO 1 I=1,N
      IF (ICONST(I).NE.0) NVAR=NVAR+1
      IF (ICONST(I).EQ.1) NC=NC+1
      IF (ICONST(I).EQ.2) NI=NI+1
1     CONTINUE
C
C          COMPUTE PHI = -PARCD(INVERSE)*PARCI
      KC=NC
      CALL SIMEQ (PARCD,X,KC,PARCDI,X)
      IF (KC.NE.NC) STOP 31
      DO 3 I=1,NC
      DO 3 J=1,NI
      SUM=0.0
      DO 2 K=1,NC
2     SUM=SUM-PARCDI(I,K)*PARCI(K,J)
3     PHI(I,J)=SUM
C
C          FILL OUT LAST ROWS OF PHI(I,J) WITH IDENTITY MATRIX.
      DO 4 I=1,NI
      DO 4 J=1,NI
      PHI(NC+I,J)=0.0
      IF (I.EQ.J) PHI(NC+I,J)=1.0
4     CONTINUE
      RETURN
      END
```

```
C     ****************************************
C     ESTIMATE CHANGES IN INITIAL VALUE GUESSES *
C     ****************************************
C
C     DELH(J) = HPAR(J,K)*HFIN(K,L)*UPDAT(L)
C
C     WHERE HPAR = A MATRIX OF PARTIAL DERIVATIVES OF THE TERMINAL
C                  CONSTRAINTS H(J), WRT. THE STATES CONSTR.(N-M+1 X KS)
C           KS   = NUMBER OF VARIABLES INVOLVED IN THE CONSTRAINTS.
C
C           HFIN = A MATRIX WHOSE FIRST N-M COLUMNS CONTAIN THE  KS
C                  FINAL VALUES OF THE CONSTRAINED VARIABLES RESULTING
C                  FROM EACH OF THE HOMOGENEOUS SOLUTIONS.
C                  THE LAST COLUMN CONTAINS THE DERIVATIVES OF THESE
C                  VARIABLES EVALUATED AT THE END POINT. (KS  X  N-M+1)
C           UPDAT= A VECTOR OF CHANGES TO BE MADE IN THE INITIAL GUESSES
C                  DELTA-T FINAL IS THE LAST ELEMENT.   (N-M+1)
C           DELH = TERMINAL BOUNDARY CONDITION DISSATISFACTIONS. (N-M+1)
C
C     IFINAL= DESCRIBES POSITION OF FINAL CONSTRAINED STATES IN X-VECTOR
C
      SUBROUTINE BOUND (X,DER,CONST,PAR,UPDAT)
      REAL X(100),DER(100),CONST(10),HPAR(10,10),HFIN(10,10),H(10,10),PA
     1R(20)
      REAL DELH(10),UPDAT(10)
      INTEGER IFINAL(10)
      REAL LU,LG,LR,M
      N=PAR(2)
      M=PAR(3)
      ITF=PAR(5)
      NTC=N-M+ITF
      DO 1 I=1,10
      DO 1 J=1,10
1     HPAR(I,J)=0.0
C
C          THE USER MUST PROVIDE SECTIONS TO EVALUATE DELH,HPAR,IFINAL.
C          THESE MUST      BE EVALUATED USING FINAL CONDITIONS.
C          FORM TERMINAL CONSTRAINT DISSATISFACTIONS.
C          VARIABLES INVOLVED IN CONSTRAINTS HAVE A 1 IN THE CORRESPOND-
C          ING POSITION OF IFINAL
      M=X(25)
      U=X(1)
      G=X(2)
      R=X(3)
      LU=X(4)
      LG=X(5)
      LR=X(6)
      DELH(1)=CONST(1)-(U**2/2.0-1.0/R)
      DELH(2)=CONST(2)-(R*U*COS(G))
      DELH(3)=CONST(3)-((U/R-1.0/(U*R**2))*LG*COS(G)+(LR*U-LU/R**2)*SIN(
     1G))
      DELH(4)=CONST(4)-(LU**2+(LG/U)**2-M**2)
C
C          FORM HPAR(J,I)
      HPAR(1,1)=U
      HPAR(1,3)=1.0/R**2
      HPAR(2,1)=R*COS(G)
      HPAR(2,2)=-R*U*SIN(G)
      HPAR(2,3)=U*COS(G)
      HPAR(3,1)=(1.0/R+1.0/((U*R)**2))*LG*COS(G)+LR*SIN(G)
      HPAR(3,2)=-(U/R-1.0/(U*R**2))*LG*SIN(G)+(LR*U-LU/R**2)*COS(G)
      HPAR(3,3)=(-U/R**2+2./(U*R**3))*LG*COS(G)+2.*LU*SIN(G)/R**3
      HPAR(3,4)=-SIN(G)/R**2
      HPAR(3,5)=(U/R-1.0/(U*R**2))*COS(G)
      HPAR(3,6)=U*SIN(G)
      HPAR(4,1)=-2.0*LG**2/U**3
      HPAR(4,4)=2.0*LU
      HPAR(4,5)=2.0*LG/U**2
C
C          FORM THE DESCRIPTION OF POSITION OF CONSTRAINT VARIABLES.
```

```
      IFINAL(1)=1
      IFINAL(2)=1
      IFINAL(3)=1
      IFINAL(4)=1
      IFINAL(5)=1
      IFINAL(6)=1
C
C          EVALUATE THE FINAL MATRIX.
      LL=N-M
      DO 2 I=1,LL
      L=0
      DO 2 K=1,N
      IF (IFINAL(K).EQ.0) GO TO 2
      L=L+1
      HFIN(L,I)=X(I*N+K)
2     CONTINUE
C
      IF (ITF.EQ.0) GO TO 4
      L=0
      DO 3 K=1,N
      IF (IFINAL(K).EQ.0) GO TO 3
      L=L+1
      HFIN(L,NTC)=DER(K)
3     CONTINUE
C
C          FORM PRODUCT
4     DO 6 J=1,NTC
      DO 6 K=1,NTC
      SUM=0.0
      DO 5 I=1,L
5     SUM=SUM+HPAR(J,I)*HFIN(I,K)
6     H(J,K)=SUM
C
C          SOLVE LINEAR EQUATIONS FOR UPDATES OF INITIAL CONDITIONS.
C          UPDAT = H(INVERSE)*DELH. (HPAR = DUMMY TO TAKE INVERSE.)
      CALL SIMEQ (H,DELH,NTC,HPAR,UPDAT)
      RETURN
      END
C     ****************************************
C     ESTIMATE CHANGES IN INITIAL VALUE GUESSES *
C     ****************************************
C
C     DELH(J) = HPAR(J,K)*HFIN(K,L)*UPDAT(L)
C
C     WHERE HPAR = A MATRIX OF PARTIAL DERIVATIVES OF THE TERMINAL
C                  CONSTRAINTS H(J), WRT. THE STATES CONSTR.(N-M+1 X KS)
C           KS   = NUMBER OF VARIABLES INVOLVED IN THE CONSTRAINTS.
C
C           HFIN = A MATRIX WHOSE FIRST N-M COLUMNS CONTAIN THE  KS
C                  FINAL VALUES OF THE CONSTRAINED VARIABLES RESULTING
C                  FRCM EACH OF THE HOMOGENEOUS SOLUTIONS.
C                  THE LAST COLUMN CONTAINS THE DERIVATIVES OF THESE
C                  VARIABLES EVALUATED AT THE END POINT. (KS  X  N-M+1)
C           UPDAT= A VECTOR OF CHANGES TO BE MADE IN THE INITIAL GUESSES
C                  DELTA-T FINAL IS THE LAST ELEMENT.   (N-M+1)
C           DELH = TERMINAL BOUNDARY CONDITION DISSATISFACTIONS. (N-M+1)
C
C     IFINAL= DESCRIBES POSITION OF FINAL CONSTRAINED STATES IN X-VECTOR
C
      SUBROUTINE FIXED (X,DER,CONST,PAR,UPDAT)
      REAL X(100),DER(100),CONST(10),HPAR(10,10),HFIN(10,10),H(10,10),PA
     1R(20)
      REAL DELH(10),UPDAT(10)
      INTEGER IFINAL(10)
      REAL LR,LU,LG,M
      N=PAR(2)
      M=PAR(3)
      ITF=PAR(5)
      NTC=N-M+ITF
      DO 1 I=1,10
      DO 1 J=1,10
1     HPAR(I,J)=0.0
C
```

```
C           THE USER MUST PROVIDE SECTIONS TO EVALUATE DELH,HPAR,IFINAL.
C           THESE MUST      BE EVALUATED USING FINAL CONDITIONS.
C           FORM TERMINAL CONSTRAINT DISSATISFACTIONS.
C           VARIABLES INVOLVED IN CONSTRAINTS HAVE A 1 IN THE CORRESPOND-
C           ING POSITION OF IFINAL
      U=X(1)
      G=X(2)
      R=X(3)
      LU=X(4)
      LG=X(5)
      LR=X(6)
      M=X(25)
      F=PAR(20)
      SQRTD=SQRT((LG/U)**2+LU**2)
      DELH(1)=CONST(1)-U
      DELH(2)=CONST(2)-G
      DELH(3)=CONST(3)-R
      DELH(4)=CONST(4)-(F*SQRTD/M+(U/R-1.0/(U*R**2))*LG*COS(G)+(LR*U-LU/
     1R**2)*SIN(G)-F)
C
C           FORM HPAR(J,I)
      HPAR(1,1)=1.0
      HPAR(2,2)=1.0
      HPAR(3,3)=1.0
      HPAR(4,1)=-F*LG**2/(SQRTD*M*U**3)+(1.0/R+1.0/((U*R)**2))*LG*COS(G)
     1+LR*SIN(G)
      HPAR(4,2)=-(U/R-1.0/(U*R**2))*LG*SIN(G)+(LR*U-LU/R**2)*COS(G)
      HPAR(4,3)=-(U/R**2-2.0/(U*R**3))*LG*COS(G)+2.0*LU*SIN(G)/R**3
      HPAR(4,4)=F*LU/(M*SQRTD)-SIN(G)/R**2
      HPAR(4,5)=F*LG/(M*SQRTD*U**2)-COS(G)*(1.0/(U*R**2)-U/R)
      HPAR(4,6)=U*SIN(G)
C
C           FORM THE DESCRIPTION OF POSITION OF CONSTRAINT VARIABLES.
      IFINAL(1)=1
      IFINAL(2)=1
      IFINAL(3)=1
      IFINAL(4)=1
      IFINAL(5)=1
      IFINAL(6)=1
C
C           EVALUATE THE FINAL MATRIX.
      LL=N-M
      DO 2 I=1,LL
      L=0
      DO 2 K=1,N
      IF (IFINAL(K).EQ.0) GO TO 2
      L=L+1
      HFIN(L,I)=X(I*N+K)
2     CONTINUE
C
      IF (ITF.EQ.0) GO TO 4
      L=0
      DO 3 K=1,N
      IF (IFINAL(K).EQ.0) GO TO 3
      L=L+1
      HFIN(L,NTC)=DER(K)
3     CONTINUE
C
C           FORM PRODUCT
4     DO 6 J=1,NTC
      DO 6 K=1,NTC
      SUM=0.0
      DO 5 I=1,L
5     SUM=SUM+HPAR(J,I)*HFIN(I,K)
6     H(J,K)=SUM
C
C           SOLVE LINEAR EQUATIONS FOR UPDATES OF INITIAL CONDITIONS.
C           UPDAT = H(INVERSE)*DELH. (HPAR = DUMMY TO TAKE INVERSE.)
      CALL SIMEQ (H,DELH,NTC,HPAR,UPDAT)
      RETURN
      END
```

```
C     *****************************************
C     SOLUTION OF SIMULTANEOUS LINEAR  EQUATIONS *
C     *****************************************
C
C          XDOT(I) = A(I,J)*X(J)
C
C     XDOT(I) = A VECTOR OF KC CONSTANTS.
C     KC      = THE NUMBER OF LINEAR EQUATIONS.
C     A(I,J)  = A KC BY KC MATRIX OF CONSTANTS.
C     SIMEQ SOLVES FOR AND RETURNS THROUGH THE CALLING LIST,
C     AINV    = A KC BY KC MATRIX = A(INVERSE)
C     X(J)    = A SOLUTION VECTOR.
C
C
      SUBROUTINE SIMEQ (A,Y,KC,AINV,X)
      DIMENSION A(10,10), B(10,10), AINV(10,10), Y(10), X(10)
C
C          SET INVERSE TO IDENTITY. SAVE A AND Y (THESE ARNT DESTROYED)
      DO 1 I=1,KC
      X(I)=Y(I)
      DO 1 J=1,KC
      B(I,J)=A(I,J)
      AINV(I,J)=0.0
      IF (I.EQ.J) AINV(I,J)=1.0
1     CONTINUE
C
C          GENERATE INVERSE AND SOLUTION SIMULTANEOUSLY BY TRANSFORMING
C          A INTO IDENTITY AND PERFORMING THE SAME OPERATIONS ON IDENT.
      DO 11 I=1,KC
C
C          FIND THE LARGEST ELEMENT IN I-TH COLUMN
      COMP=0.0
      DO 3 K=I,KC
      TEMP=B(K,I)
      IF (ABS(TEMP).GT.ABS(COMP)) 2,3
2     COMP=TEMP
      N=K
3     CONTINUE
C
C          IF LARGEST ELEMENT IS ZERO, THEN MATRIX IS SINGULAR.
      IF (COMP.EQ.0.0) 4,6
4     WRITE (3,5) KC,KC
    5 FORMAT(//// 20X, *LINEAR SIMULTANEOUS EQUATIONS SUBR. --- SINGULAR
     2 *, I2, * BY *, I2, * MATRIX.*//)
      KC=-KC
      RETURN
C
C          CHECK FOR THE LARGEST ELEMENT ON THE DIAGONAL
C          IF NOT ON DIAGONAL, INTERCHANGE COLUMNS I AND N.
6     IF (N.EQ.I) GO TO 8
      TEMP=X(I)
      X(I)=X(N)
      X(N)=TEMP
      DO 7 M=1,KC
      TEMP=B(I,M)
      B(I,M)=B(N,M)
      B(N,M)=TEMP
      TEMP=AINV(I,M)
      AINV(I,M)=AINV(N,M)
7     AINV(N,M)=TEMP
C
```

```
C         LARGEST ELEMENT IS NOW ON THE DIAGONAL.
C         DIVID THROUGH COLUMN BY THE DIAGONAL ELEMENT.
8     TEMP=1.0/B(I,I)
      X(I)=X(I)*TEMP
      DO 9 M=1,KC
      B(I,M)=B(I,M)*TEMP
9     AINV(I,M)=AINV(I,M)*TEMP
C
C         DIAGONALIZE B THUS GENERATING AINV AND X.
      DO 11 J=1,KC
      TEMP=B(J,I)
      IF (I.EQ.J.OR.TEMP.EQ.0.0) GO TO 11
      X(J)=X(J)-TEMP*X(I)
      DO 10 N=1,KC
      B(J,N)=B(J,N)-TEMP*B(I,N)
10    AINV(J,N)=AINV(J,N)-TEMP*AINV(I,N)
11    CONTINUE
      RETURN
      END
```

REFERENCES

Bell, D. J. "Optimal Space Trajectories - A Review of Published Work," The Aeronautical Journal of the Royal Aeronautical Society, Vol. 72, No. 2 (1968), pp. 141-6.

Bellman, R. E., and Kalba, R. E. Quasilinearization and Nonlinear Boundary Value Problems, American Elsevier, New York (1965).

Bliss, G. A. Transactions of the American Mathematical Society, Vol. 3 (1902), p. 132.

Bliss, G. A. Lectures on the Calculus of Variations, University of Chicago Press, Chicago (1946).

Bolza, O. "Uber den anormalen Fall beim Lagrangeschen und Mayerschen Problem mit gemischten Bedingungen und variabeln Endpunkten," Mathematische Annalen, LXXIV (1913), pp. 430-46.

Bolza, O. Lectures on the Calculus of Variations, Dover Publications, New York (1961); also, University of Chicago Press, Chicago (1904).

Bryson, A. E., and Ho, Y. C. Optimization Estimation, and Control, Blaisdell,New York (1969).

Buck, R. C. Advanced Calculus, McGraw-Hill Book Co., New York (1965).

Clebsch, A. Crelle's Journal, tlv (1858), pp. 254-73 and pp. 335-55.

Dienger, J. Grundriss der Variationsrechnung, (1867).

Erdmann, B. Zeitschrift für Mathematik und Physik, Vol. XXIII (1878), p. 364.

Euler, L. Methodus inveniendi lineas curvas miximi minimive proprietate gaudentes, sive solutio problematis isoperimetrici lattissimo sensu accepti, Lausanne et Geneva, (1744); translated in: Ostwald, Klassiker der exacten Wissenschaften, No. 46, Lipsiae (1911).

Forsyth, A. R. Calculus of Variations, Cambridge University Press, Cambridge (1927); also Dover Publications, New York (1960).

Handelsman, M. "Optimal Free Space Trajectories Using Impulsive Trajectories as Starting Interatives" American Institute of Aeronautics and Astronautics Journal, Vol. 4, No. 6 (1966), pp. 1077-82.

Hestenes, M. R. Numerical Methods of Obtaining Solutions of Fixed Endpoint Problems in the Calculus of Variations, RM-102, The Rand Corporation (1949).

Hestenes, M. R. Calculus of Variations and Optimal Control Theory, Wiley & Sons, New York (1966).

Hilbert, D. Bull. of the American Mathematical Society, Vol. VIII (1902), translated by Mrs. Newton, p. 121.

Hildebrand, F. B. Advanced Calculus for Applications, Prentice-Hall, Englewood Cliffs, N. J. (1948).

Hillsley, R. H., and Robbins, H. M. "A Steepest Ascent Trajectory Optimization Method Which Reduces Memory Requirements," Computating Methods in Optimization Problems, edited by Balkrishnan, A. V., and Neustadt, L. W., Academic Press, New York (1964).

Householder, A. S. "The Dependency of a Focal Point Upon Curvature in the Calculus of Variations," Contributions to the Calculus of Variations, University of Chicago Press, Chicago (1937), pp. 485-526.

Jacobi, K. Crelle's Journal, Vol. txvii (1837), pp. 68-82.

Johnson, D. C., and Gibson, J. E. "Singular Solutions in Problems of Optimal Control," IEEE Transactions on Automatic Control, Vol. AC-8 (1963), pp. 4-14.

Kalman, R. E. "A Theory of Difficulty of Computation in Optimal Control," Scientific Computation Symposium: Control Theory and Applications. Thomas J. Watson Research Center, Yorktown Heights, New York (1964).

Kelly, H. J. "Method of Gradients," Optimization Techniques, Edited by Leitmann, G., Academic Press, New York (1962).

Kelly, H. J., Kopp, R. E., and Moyer, H. G. "A Trajectory Optimization Technique Based upon the Theory of the Second Variation," Progress in Astronautics and Aeronautics: Celestial Mechanics and Astrodynamics, edited by Szebehely, V. G., Vol. 14, Academic Press, New York (1964).

Kelly, H. J., Kopp, R. E., and Moyer, H. G. "Singular Extremals," Topics in Optimization, edited by Leitmann, G., Academic Press, New York (1967).

Kneser, A. Lehrubuch der Variationsrechnung, Braunschweig (1900).

Kopp, R. E., and Moyer, H. G. "Trajectory Optimization Techniques," Advances in Control Systems, edited by Leondes, C. T., Vol. 4, Chapter 3, Academic Press, New York (1966).

Lagrange, J. L. Oeuvres de Lagrange, Gauthier-Villars, Paris (1867).

Lawden, D. F. Optimal Trajectories for Space Navigation, Butterworth, London (1963).

Legendre, A. Mémoires de l'Acadámie Royal des Sciences, (1786), pp. 7-37.

Leitmann, G. An Introduction to Optimal Control, McGraw-Hill Book Co., New York (1966), pp. 57-8.

Lewallen, J. M. "A Modified Quasilinearization Method for Solving Trajectory Optimization Problems," American Institute of Aeronautics and Astronautics Journal, Vol. 5, No. 5 (1967), pp. 962-6.

Lewallen, J. M., Tapley, B. D., and Williams, S. D. "Iteration Procedures for Indirect Trajectory Optimization," Journal of Spacecraft and Rockets, Vol. 5, No. 3 (1968), pp. 321-7.

Long, R. S. "Newton-Raphson Operator; Problems with Undetermined Endpoints," American Institute of Aeronautics and Astronautics Journal, Vol. 3, No. 7 (1965), pp. 1351-2.

Mason, J. D. *Transformations and Discontinuities For Optimal Space Trajectories*, Ph. D. dissertation, University of Arizona (1967).

Mayer, A. "Beitrage zur Theirie der Maxima einfacher Integral," *Crelle's Journal*, tlxix (1868), pp. 238-63.

Mayer, A. "Zur Aufstellung der Kriterien des Maximums and Minimums der einfachen Integrale bei variabeln Grenswerten," *Leipziger Berichte*, XXXVI (1884), pp. 99-128.

Mayer, A. "Zur Aufstellung der Driterien des Maximums und Minimums der einfachen Integrale bei variabeln Grenswerten," *Leipziger Berichte*, XLVIII (1896), pp. 436-65.

McCue, G. A. Quasilinearization Determination of Optimum Finite-Thrust Orbital Transfers," *American Institute of Aeronautics and Astronautics Journal*, Vol. 5, No. 4 (1967), pp. 755-63.

McCue, G. A., and Bender, D. F. "Numerical Investigation of Minimum Impulse Orbital Transfer," *American Institute of Aeronautics and Astronautics Journal*, Vol. 3, No. 3 (1965), pp. 2328-34.

McGill, R., and Kenneth, P. "A Convergence Theorem on Iterative Solution of Nonlinear Two-point Boundary Value Systems," XIV-th International Astronautical Federation Congress, Paris (1963).

McGill, R., and Kenneth, P. "Solution of Variational Problems by Means of Generalized Newton-Raphson Operator," *American Institute of Aeronautics and Astronautics Journal*, Vol. 2, No. 10 (1964), pp. 1761-6.

McIntyre, J. E., and Crocco, L. "Linearized Treatment of the Optimal Transfer of a Thrust Limited Vehicle Between Coplanar Circular Orbits," *Astronautica Acta*, Vol. 12, No. 4 (1966), pp. 224-34.

McIntyre, J. E., and Crocco, L. "Higher Order Treatment of the Orbital Transfer of a Thrust-Limited Vehicle between Coplanar Circular Orbits," *Astronautica Acta*, Vol. 13, No. 1 (1967), pp. 3-21.

Moyer, H. G., and Pinkham, G. P. "Several Trajectory Optimization Techniques," *Computing Methods in Optimization Problems*, edited by Balakrishnan, A. V., and Neustadt, L. W., Academic Press, New York (1964).

Ostwald, W. *Klassiker der exacten Wissenshaften*, Lipsiae (1911).

Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko, E. F. *The Mathematical Theory of Optimal Processes*, edited by Neustadt, L. W., translated by Trirogoff, K. M., Wiley & Sons, New York (1962).

Robbins, H. M. "Optimality of Intermediate Thrust Arcs of Rocket Trajectories," *American Institute of Aeronautics and Astronautics Journal*, Vol. 3, No. 6 (1965), pp. 1094-8.

Vincent, T. L., Unpublished course notes for "Performance Optimization in Aerospace Engineering," AME 367, University of Arizona, Tucson, Arizona (1969).

Vincent, T. L., and Brusch, R. G. "Minimum Time Aircraft Trajectories between Two Points in Range Altitude Space," NASA CR-631, National Aeronautics and Space Administration, Washington, D. C. (1966).

Vincent, T. L., and Mason, J. D. "Discontinuous Optimal Trajectories," *Journal of Optimization and Applications*, Vol. 3, No. 4 (1969), pp. 263-81.